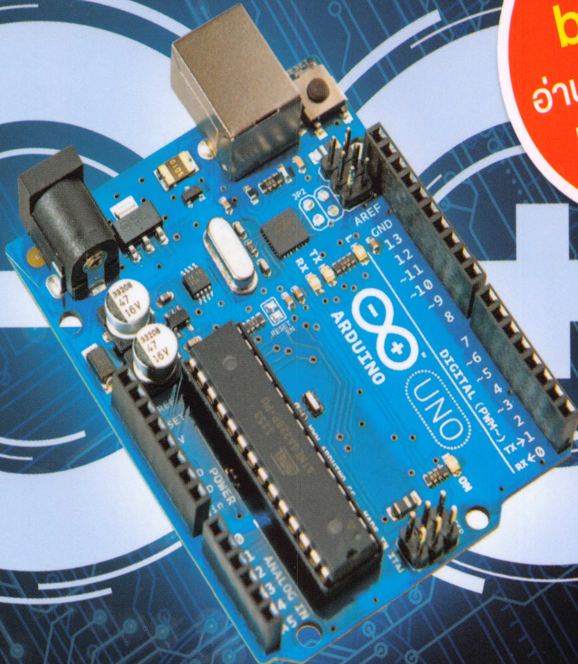
CORE FUNCTION

ARDUINO STARTUP สนุกสุดเหวี่ยงกับเซ็นเซอร์

Step by Step
อ่านง่าย เข้าใจง่าย เรียนรู้ได้ด้วยตนเอง



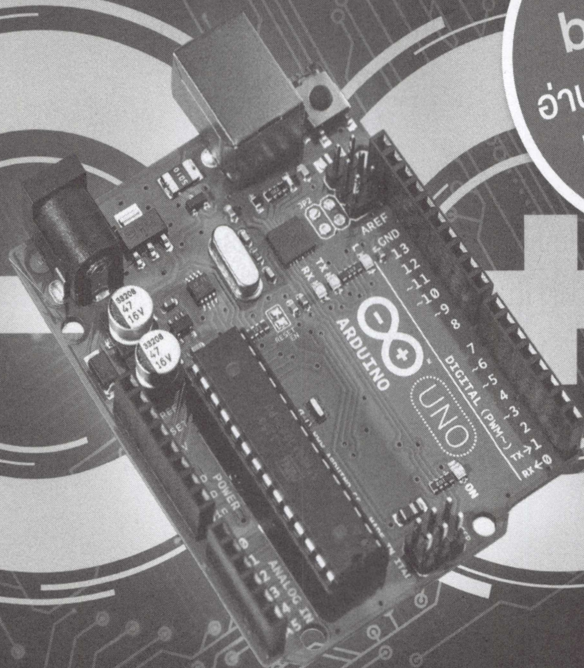
- ทำความเข้าใจกับบอร์ด Arduino และซอฟต์แวร์ทั้งหมดที่จำเป็น
- ทดลองและเล่นกับเซ็นเซอร์หลากหลายแบบที่มีในตลาด พร้อมโปรแกรมสั่งงานและการต่อวงจร
- เปลี่ยนความคิด เพราะ Arduino ไม่ใช่ของยาก

อ.นพ มหิษานนท์

CORE FUNCTION

ARDUINO STARTUP สนุกสุดเหวี่ยงกับเซ็นเซอร์

Step by Step
อ่านง่าย เข้าใจง่าย
เรียนรู้ได้ด้วยตนเอง



+ ทำความเข้าใจกับบอร์ด Arduino และซอฟต์แวร์ทั้งหมดที่จำเป็น

− ทดลองและเล่นกับเซ็นเซอร์หลากหลายแบบที่มีในตลาด พร้อมโปรแกรมสั่งงานและการต่อวงจร

+ เปลี่ยนความคิด เพราะ Arduino ไม่ใช่ของยาก

อ.นพ มหิษานนท์

Arduino Startup
สนุกสุดเหวี่ยงกับเซ็นเซอร์

ผู้เขียน : อ.นพ มหิษานนท์

บรรณาธิการ : จิระ จริงจิตร

ออกแบบปก : ธันวา เรืองสวัสดิ์

เลขมาตรฐานสากลประจำหนังสือ : 978-616-7502-84-7

พิมพ์ครั้งที่ 1 : ตุลาคม 2561

ราคา 265 บาท

จัดพิมพ์โดย สำนักพิมพ์ คอร์ฟังก์ชั่น

แยกสี/เพลท : Classic Scan

โทร. 0-2291-7575

พิมพ์ : บริษัท ส.เอเซียเพลส (1989) จำกัด

โทร. 0-2732-3101-6

จัดจำหน่ายโดย : บริษัท เอ-บุ๊ค ดิสทริบิวชั่น จำกัด

โทร. 0-2968-9337, 0-2968-9207

คำนำ

น้องๆ ส่วนใหญ่จะเริ่มสนใจกับบอร์ดไมโครคอนโทรลเลอร์ หลังจากที่หมดสนุกจากชุดคิดส์อิเล็กทรอนิกส์แล้ว เพราะไมโครคอนโทรลเลอร์เป็นอะไรที่เหนือกว่า พิเศษกว่า และพลิกแพลง ปรับเปลี่ยน พัฒนาได้มากกว่า “เท่าที่ต้องการ”

แต่ก็มีน้องๆ หลายคนที่หมดหวังท้อแท้กับการใช้งานบอร์ดไมโครคอนโทรลเลอร์ เพราะไม่เข้าใจภาษา C สำหรับไมโครคอนโทรลเลอร์ เขียนโปรแกรมไม่เป็น เขียนโปรแกรมไม่สำเร็จ ชุดโมดูลอิเล็กทรอนิกส์ไม่ทำงานตามที่ตั้งใจเอาไว้ และยังมีคนอีกมากมายที่ทิ้งบอร์ดไมโครคอนโทรลเลอร์ไว้ในกล่องบนชั้นเก็บของแบบถาวร เพราะไม่รู้เรื่องการเขียนโปรแกรม จึงจำเป็นต้องล้มเลิกโครงการไมโครคอนโทรลเลอร์ทั้งหมด ทั้งที่ยังไม่เคยเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์กับคอมพิวเตอร์เลยด้วยซ้ำ หากคุณกำลังมีอาการท้อแท้เช่นว่านี้ แสดงว่า คุณกำลังเรียนรู้และทำความเข้าใจกับทุกสิ่งทุกอย่างของบอร์ดไมโครคอนโทรลเลอร์อย่างผิดวิธี

หนังสือเล่มนี้แนะนำการทำความเข้าใจกับบอร์ดไมโครคอนโทรลเลอร์ และ Software ทั้งหมดของบอร์ดไมโครคอนโทรลเลอร์ โดยยกตัวอย่างบอร์ดไมโครคอนโทรลเลอร์ที่ได้รับความนิยมสูงสุดในขณะนี้ Arduino เริ่มตั้งแต่ขั้นตอนการแกะกล่อง Arduino ไปจนถึงการต่อวงจร และสร้างโปรแกรมให้บอร์ด Arduino ส่งสัญญาณคำสั่งให้โมดูลอิเล็กทรอนิกส์ทำงาน พร้อมด้วยโครงงาน Arduino มากกว่า 30 โครงงานที่อัดแน่นอยู่ในหนังสือเล่มนี้

ถ้าการเรียนรู้วิธีใช้งานบอร์ด Arduino เป็นไปอย่างสนุกสุดมันส์ คุณจะสามารถสนุกสุดเหวี่ยงกับบอร์ด Arduino ได้ และอยากจะศึกษาเรียนรู้แบบไม่รู้จบ เมื่อเปิดหนังสือเล่มนี้แล้ว คุณจะถูกเปลี่ยนความคิดและวิธีการเข้าใจกับการใช้งานบอร์ด Arduino ไปโดยสิ้นเชิง แล้วคุณจะพบกับความสนุกสุดเหวี่ยงที่เกิดขึ้นเมื่อกด Upload Sketch Code ไปยังบอร์ด Arduino

สำนักพิมพ์ Core Function

สารบัญ



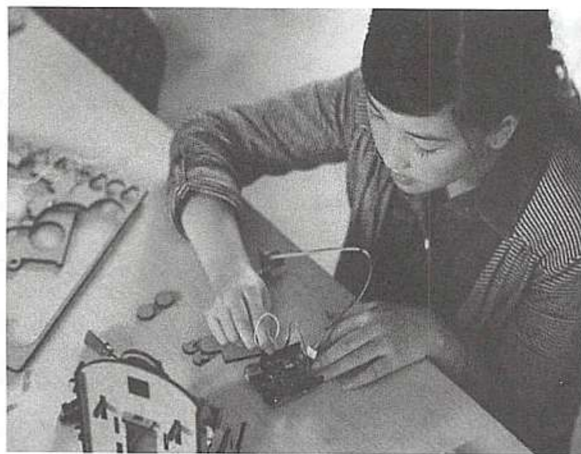
1

- ไมโครคอนโทรลเลอร์ คืออะไร
- ไมโครคอนโทรลเลอร์แบบ Open-Source Platform
- โปรแกรม Arduino IDE
- Arduino Web Editor

ไมโครคอนโทรลเลอร์ และ Arduino คืออะไร

"ระบบการเรียนรู้เปลี่ยนไปแล้ว" กับโลกยุคใหม่ที่มีเครื่องอำนวยความสะดวกในการศึกษาเรียนรู้ทุกศาสตร์ทุกแขนง ถ้าเรากำลังพูดถึง "ไมโครคอนโทรลเลอร์" หากย้อนเวลากลับไปสัก 20 ปีก่อน สมมุติว่าเราอยากจะลองซื้อบอร์ดไมโครคอนโทรเลอร์มาใช้งาน ต่อสายนั่นสายนี่ ลองต่อเข้ากับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ "ประมาณว่ามีประสบการณ์กับการประกอบชุดคิดส์มาแล้ว" กำแพงแห่งความคิดอย่างหนึ่งจะก่อขึ้นขวางกั้นการเขาถึงบอร์ดไมโครคอนโทรลเลอร์ของเราทันที กำแพงคำถามนั้น คือ "รู้จักภาษา C หรือไม่ เขียนภาษา C เป็นมั้ย แล้วภาษา C คืออะไร" เมื่อคำถามนี้เกิดขึ้นในหัวเราเมื่อไหร่ ไม่ว่าจะเป็นเพราะมีคนถาม หรือศึกษารายละเอียดแล้วเข้าใจได้เอง เมื่อนั้นความท้อแท้หมดกำลังใจกำลังจะเกิดขึ้น เพราะเมื่อลองไปดูหนังสือ "การเขียนภาษา C" เราจะพบว่า "มันยากมาก เปิดผ่านตาไป 20 หน้า ก็ยังไม่รู้เรื่องอะไรเลยเกี่ยวกับภาษา C" บทสรุปการเข้าถึงบอร์ดไมโครคอนโทรลเลอร์นี้ก็คือ "ตกม้าตายตั้งแต่ยังไม่เห็นตัวม้าเลยด้วยซ้ำ" แค่เราคิดว่าภาษา C ที่ใช้ในการเขียนโปรแกรมเพื่อใช้งานไมโครคอนโทรลเลอร์ "มันหินจนมองไม่เห็นอนาคต" การเชื่อมโยงสู่เรียนรู้ก็ไม่มีทางเกิดขึ้น "เพราะยังไม่ได้สัมผัสกับบอร์ดไมโครคอนโทรลเลอร์ของจริงเลยด้วยซ้ำ"

สำหรับทุกวันนี้ ระบบการศึกษาไทยเข้าใจได้เป็นอย่างดีแล้วว่า การศึกษาเรียนรู้ที่ได้ผลสัมฤทธิ์ดีที่สุด คือ การลงมือทำด้วยตัวเอง เพื่อสร้างเสริมประสบการณ์ในศาสตร์วิชาที่อยากเรียนรู้ด้วยตนเอง การศึกษาเรียนรู้ที่ได้ผลสัมฤทธิ์ต่ำ คือ การพยายามทำความเข้าใจเพียงเนื้อหาในตำราเรียน ทั้งระบบการศึกษาไทยและคนไทยเกือบจะทุกคน เข้าใจดีแล้วว่า ทฤษฎีและหลักการซึ่งเป็นตัวหนังสือและตัวเลขสูตรคำนวณ เป็นเพียงข้อมูลอ้างอิงหลังจากที่ผู้เรียนได้ลงมือทำไปแล้ว ในอดีตการศึกษาไทยมีระบบการเรียนรู้ในลักษณะ “เรียนในตำราให้เข้าใจก่อน แล้วค่อยลงมือทำ” เช่นนั้นเด็กเรียนเก่ง จึงได้เปรียบเด็กเรียนไม่เก่งไปจนถึงเด็กเรียนปานกลางมาโดยตลอด



ยกตัวอย่างการเรียนภาษาอังกฤษ สมัยก่อนเด็กๆ (เราตอนเด็กๆ นั่นแหละ) จะต้องหมกมุ่นอยู่กับเรื่อง Verbs ทั้ง 3 ช่อง และคำศัพท์หลายร้อยคำที่ครูบอกว่า “จะออกข้อสอบนะ” เราในสมัยก่อนมุ่งเน้นแต่เรื่อง Verbs กับคำศัพท์อย่างเอาเป็นเอาตาย แต่ในขณะเดียวกันหญิงไทยที่ได้เป็นภรรยาชาวต่างชาติ (อเมริกันและอังกฤษ) พูดฝรั่งได้คล่องปร๋อ “สำเนียงยังกะเจ้าของภาษา” ทั้งๆ ที่พวกเธอไม่ได้เข้าห้องเรียนภาษาอังกฤษ ไม่ได้ท่องกิริยา 3 ช่อง ไม่ต้องท่องคำศัพท์อย่างเอาเป็นเอาตาย “เพื่อจะได้สอบปลายภาคให้ได้คะแนนดีที่สุด หรือบางคนหวังแค่ผ่าน”

ในขณะที่เด็ก ม.5 ท่องศัพท์ภาษาอังกฤษแก้มน้ำตา abjure, abrogate, abstemious ภรรยาชาวต่างชาติที่เป็นคนไทย สามารถสมัครเข้าทำงานเป็น Receptionist ของโรงแรม แต่เด็ก ม.5 ที่นั่งท่องศัพท์เกือบตาย เพื่อหวังให้ได้คะแนนดีๆ จะได้มีโอกาสที่ดีในการเข้าเรียนต่อในระดับมหาวิทยาลัย ท้ายที่สุดได้ทำงานเป็นข้าราชการแผนกการเงินและบัญชี แล้วพวกเธอก็ได้รู้แล้วว่า การท่องศัพท์แทบเป็นแทบตายเมื่อ 5-6 ปีก่อน "ไม่ได้มีประโยชน์อะไรกับอาชีพฉันเลย" บางคนจบเอกภาษาอังกฤษ ได้ทำงานเป็นหัวหน้าฝ่ายต้อนรับของโรงแรม "เป็นหัวหน้า Receptionist ที่มีภรรยาเป็นฝรั่ง" หัวหน้าแผนกพูดภาษาอังกฤษกับลูกค้าโรงแรมแทบไม่รู้เรื่อง เพียงแต่รู้ว่าลูกค้าต้องการอะไรเท่านั้น "เพราะแม่นแต่คำศัพท์" เรียงประโยคตอบลูกค้าไม่เป็น "สู้ลูกน้องไม่ได้" พูดสนทนาโต้ตอบกับลูกค้าได้เป็นอย่างดี "เพราะมีภรรยาเป็นฝรั่ง" ลองคิดดูว่าผู้จัดการโรงแรมจะคิดแบบไหน แล้วหัวหน้าแผนกคนนี้จะคิดอย่างไร "รู้งี้เอาเมียฝรั่งซะก็สิ้นเรื่อง"

เอาง่ายๆ อย่างเราพูดและเขียนภาษาไทย เราเริ่มจากการพูด "พ่อ แม่ หนม ข้าว หมา แมว ฯลฯ" ยิ่งโตมากเท่าไหร่ เรายิ่งพูดชัดและรู้คำศัพท์ได้มากขึ้นเท่านั้น จนเมื่อเราได้เข้าโรงเรียน เราถูกสอนให้รู้จักตัวพยัญชนะ ตัวสระ เราจึงได้รู้ว่า พ่อ เขียนอย่างไร แม่ เขียนอย่างไร คำว่า ขนม ทำไม นม ถึงออกเสียง เป็น หนม ตามธรรมชาติเราจะต้องพูดได้อย่างเข้าใจก่อน จึงจะสามารถเขียนได้อย่างเข้าใจ "โดยไม่ต้องใช้วิธีท่องจำมากนัก" เพราะพูดด้วยตัวเอง สร้างประสบการณ์ด้วยตัวเอง "ไม่ใช่หวังพึ่งแต่ประสบการณ์ของคนอื่นที่บรรจุอยู่ในตำราเรียน" เช่นนั้นการเรียนภาษาอังกฤษที่มีผลสัมฤทธิ์ดีที่สุด คือ การหัดพูดให้เข้าใจก่อน แล้วค่อยไปทำความเข้าใจในเรื่องการเขียน และคำศัพท์ที่จำเป็นต้องจำ

หากนำเอามาเปรียบเทียบกับการเรียนรู้เรื่องการทำงานของ "ไมโครคอนโทรลเลอร์" ถ้าเราจะมัวหมกมุ่นอยู่แต่กับการเขียนภาษา C สมมุติว่ามีคนสนใจอยากเรียนรู้เรื่องบอร์ดไมโครคอนโทรลเลอร์สัก 100 คน อาจจะเหลือไม่ถึง 20 คน ที่สามารถใช้บอร์ดไมโครคอนโทรลเลอร์ได้ "เพราะภาษา C มันยาก" แต่ถ้าเราเปลี่ยนแนวทางในการเรียนรู้เสียใหม่ "ยังไม่ต้องสนใจกับการเขียนโปรแกรมตอนนี้" ซื้อบอร์ดไมโครคอนโทรลเลอร์มาเลย "แล้วลองลงมือทำด้วย

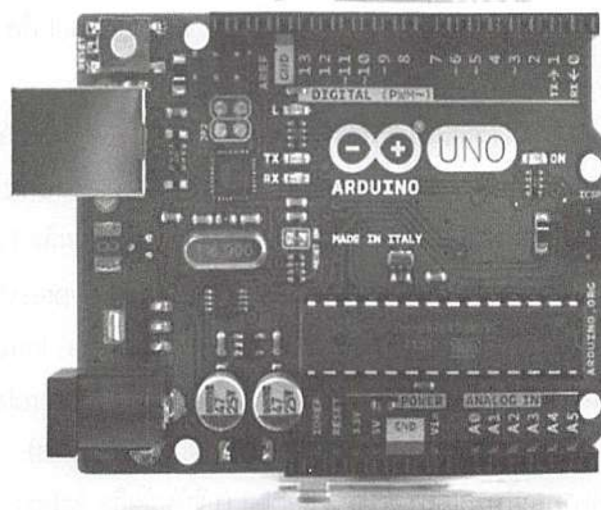
ตนเอง" ดูหนังสือคู่มือที่แถมมากับบอร์ด ทำความรู้จักและเรียนรู้หลักการทำงานและหลักการใช้งานของอุปกรณ์อิเล็กทรอนิกส์ต่างๆ จากสื่อการเรียนรู้มากมายในอินเตอร์เน็ต เช่น เว็บบอร์ด YouTube หรือ Arduino Project Hub เมื่อเราทดลองเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์กับระบบปฏิบัติการคอมพิวเตอร์ ลองเชื่อมต่อบอร์ดกับอุปกรณ์อิเล็กทรอนิกส์ ลองเขียนโปรแกรมด้วยภาษา C ลงบนโปรแกรม IDE ตามหนังสือคู่มือ เพื่อสั่งการให้บอร์ดไมโครคอนโทรลเลอร์ทำงาน เราจะรู้ได้ในทันทีว่า ทำไมต้องพิมพ์ภาษา C ประโยคนี้ "digitalWrite(3,HIGH);" เพราะเมื่อเราดูลักษณะการต่อสายจากบอร์ดไปอุปกรณ์แล้ว "ให้ขา 3 เป็นลอจิก 1" เราจะเข้าใจทันทีจากการลงมือทำด้วยตัวเอง ดีกว่าจะไปนั่งจินตนาการเองกับศึกษาประสบการณ์ของคนอื่น จากรูปภาพเพียงเล็กน้อยบนตำราสอนเขียนโปรแกรมภาษา C "แล้วเมื่อไหร่กันล่ะ"

ส่วนใหญ่คนที่สนใจบอร์ด Arduino จะท้อแท้หมดหวังตั้งแต่เริ่มเรียนรู้การใช้ภาษา C สรุปจึงไม่ได้ลิ้มลองกับความสนุกที่เกิดขึ้นกับการใช้บอร์ด Arduino จริงๆ เลยแม้แต่วินาทีเดียว เพราะพลังชีวิตทั้งหมดถูกทุ่มเทไปกับการทำความเข้าใจกับหลักการเขียนโปรแกรมไปสิ้นแล้ว "ไร้ประโยชน์สิ้นดี"



ไมโครคอนโทรลเลอร์ คืออะไร

ไมโครคอนโทรลเลอร์ คือ ชุดควบคุมขนาดเล็กที่รวบรวมคุณสมบัติหลักของระบบคอมพิวเตอร์ มารวมไว้ในตัวถังเดียวกัน "ในบอร์ดเดียวกัน" ไม่ว่าจะเป็นส่วนประกอบหลักอย่าง CPU หน่วยความจำ และพอร์ตเชื่อมต่อกับอุปกรณ์ต่างๆ ทั้งหมดรวมอยู่ในชุดเดียวกัน แต่คุณสมบัติหลักๆ ของไมโครคอนโทรลเลอร์ที่ดูเหมือนจะเป็นคุณสมบัติเฉพาะตัว ซึ่งเป็นความเก่งคนละด้านต่างจากคอมพิวเตอร์ทั่วไป "แต่เป็นระบบเดียวกัน" คือ มีพอร์ตติดต่อกับอุปกรณ์ภายนอก และอุปกรณ์ภายนอกเหล่านั้นไม่ใช่เพียงคีย์บอร์ด เมาส์ แฟรชไดรว์ หรือจอมอนิเตอร์ แต่อุปกรณ์ที่กำลังจะกล่าวถึงนี้ คือ อุปกรณ์ภายนอกที่เป็นชิ้นส่วนอิเล็กทรอนิกส์ หรือชุดอุปกรณ์อิเล็กทรอนิกส์ที่ถูกประดิษฐ์ขึ้นมาเพื่อใช้งานในลักษณะเฉพาะ (โมดูล) เช่น โมดูลวัดค่า โมดูลสื่อสาร โมดูลการบันทึกค่าและส่งค่า การทำงานของมอเตอร์ การทำงานของรีเลย์ การแสดงแสง สี เสียง และจอลดทอนสัญญาณรบกวน และ GPS ไมโครคอนโทรลเลอร์ จะเป็นอุปกรณ์ควบคุมการทำงานของอุปกรณ์ภายนอกเหล่านี้ โดยส่งสัญญาณคำสั่งและรับสัญญาณค่าต่างๆ จากอุปกรณ์ภายนอกผ่านพอร์ต ที่ถูกติดตั้งเชื่อมโยงกับระบบการประมวลผล กล่าวคือ ไมโครคอนโทรลเลอร์จะส่งสัญญาณคำสั่งผ่านพอร์ตอินพุต (Input



Port) ไปยังอุปกรณ์ภายนอก และรับสัญญาณค่าต่างๆ จากอุปกรณ์ภายนอกผ่านพอร์ตเอาต์พุต (Output Port) แล้วนำสัญญาณที่ได้รับไปทำการประมวลผล

แน่นอนไม่ว่าเครื่องใช้ไฟฟ้าชนิดใดแบบไหน หากจะต้องมีการควบคุมความเร็ว หรือควบคุมการแสดงผล และควบคุมปริมาณ จะต้องมีไมโครคอนโทรลเลอร์เป็นอุปกรณ์ควบคุมการทำงานของเครื่องใช้ไฟฟ้า โดยจะมีการเขียนโปรแกรมคำสั่งบันทึกลงไปใน CPU ของไมโครคอนโทรลเลอร์ เพื่อส่งสัญญาณคำสั่งให้อุปกรณ์ในเครื่องใช้ไฟฟ้าทำงาน หรือรับสัญญาณค่าต่างๆ ที่อุปกรณ์ภายนอกวัดได้ เพื่อนำสัญญาณค่าเหล่านั้นไปทำการประมวลผล

แม้แต่ไฟฉายขนาดเท่านิ้วหัวแม่มือ ยังจะต้องมีบอร์ดไมโครคอนโทรลเลอร์ควบคุมการทำงาน เช่น ควบคุมระบบชาร์จแบตเตอรี่ ควบคุมโหมดแสงสว่าง ควบคุมโหมดสวิตช์ เป็นต้น

แต่ไม่ได้หมายถึง ไมโครคอนโทรลเลอร์ คือ Arduino เพียงอย่างเดียว เพราะไมโครคอนโทรลเลอร์มีหลายรุ่น หลายยี่ห้อ หลายผู้ผลิต แต่ละรุ่นอาจเลือกใช้ PCU ที่แตกต่างกัน ตามคุณสมบัติ หรือตามต้นทุนการผลิต ตามเทคโนโลยีที่ค้นพบ และตามแบบฉบับสถาปัตยกรรมที่มีการจดลิขสิทธิ์

1. ไมโครคอนโทรลเลอร์ ตระกูล PIC (บริษัทผู้ผลิต Microchip)
2. ไมโครคอนโทรลเลอร์ ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
3. ไมโครคอนโทรลเลอร์ ตระกูล AVR (บริษัทผู้ผลิต Atmel คือ Arduino พระเอกของเรานี่เอง)
4. ไมโครคอนโทรลเลอร์ ตระกูล ARM7, ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Samsung, STMicroelectronics)
5. ไมโครคอนโทรลเลอร์ ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
6. ไมโครคอนโทรลเลอร์ ตระกูล PSOC (บริษัทผู้ผลิต Cypress)
7. ไมโครคอนโทรลเลอร์ ตระกูล MSP (บริษัทผู้ผลิต Texas Intruments)
8. ไมโครคอนโทรลเลอร์ ตระกูล 68HC (บริษัทผู้ผลิต Motorola)
9. ไมโครคอนโทรลเลอร์ ตระกูล H8 (บริษัทผู้ผลิต Renesas)
10. ไมโครคอนโทรลเลอร์ ตระกูล Rabbit (บริษัทผู้ผลิต Rabbit

Semiconductor)

11. ไมโครคอนโทรลเลอร์ ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

 Arduino คืออะไร

Arduino เป็นภาษาอิตาเลี่ยน อ่านว่า อาดุอีโน่ จนเพี้ยนมาเป็น อาร์ดุยโน่ ในภาษาอื่นๆ โดย Arduino คือ ไมโครคอนโทรลเลอร์แบบ Open-Source Platform สำหรับการสร้างต้นแบบโปรแกรมการทำงานของวงจร ซึ่งมีส่วนประกอบเป็นอุปกรณ์อิเล็กทรอนิกส์ แน่นอนเครื่องใช้ไฟฟ้าแทบจะทุกชนิด ตั้งแต่ไม้ซ็อตยุงไปจนถึงเครื่องคอมพิวเตอร์ ต่างก็ต้องมีบอร์ดไมโครคอนโทรลเลอร์ที่ถูก Upload โปรแกรมการทำงานตามคุณสมบัติที่ผู้ออกแบบต้องการ แต่คำว่า "ไมโครคอนโทรลเลอร์แบบ Open-Source Platform" เป็นความพิเศษที่เป็นข้อดีของบอร์ด Arduino โดยเปิดกว้างให้ผู้คนสามารถพัฒนาและประยุกต์ใช้งานอย่างเสรี ไม่มีลิขสิทธิ์ จึงทำให้บอร์ด Arduino เป็นที่รู้จักและนิยมนำมาใช้งานกันอย่างแพร่หลาย

และที่สำคัญ Platform ของไมโครคอนโทรลเลอร์อย่าง Arduino ง่ายต่อการใช้งาน ซึ่งถือเป็นจุดมุ่งหมายของการสร้างไมโครคอนโทรลเลอร์ สำหรับผู้คนทั่วโลก "โดยเฉพาะเด็กๆ ที่มีความสนใจในเทคโนโลยี" ทำให้บอร์ด Arduino ถูกพัฒนาด้วยความร่วมมือจากผู้คนทั่วโลก และจำหน่ายในราคาที่เด็กๆ สามารถจับต้องได้ไม่ยากเท่าใดนัก

ข้อดีอีกข้อหนึ่งที่สำคัญของบอร์ด Arduino คือ เป็นบอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ขนาดพอๆ กับบัตรประชาชน (หรืออาจจะเล็กกว่าด้วยซ้ำ) โดยส่วนประกอบหลักของบอร์ด คือ ไมโครคอนโทรลเลอร์ และนำมาประกอบรวมกันกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ และถูกพัฒนาให้ง่ายต่อการใช้งานมากขึ้นเรื่อยๆ ซึ่งทั้งนี้บอร์ด Arduino จะมีหลายรุ่นให้เลือกใช้ โดยแต่ละรุ่นของ Arduino อาจ



มีความแตกต่างกัน เริ่มจากขนาดของบอร์ด หรือสเปกซึ่งเป็นคุณสมบัติพิเศษอย่างอื่น เช่น จำนวนของ Pin แรงดันไฟฟ้าที่ใช้ แรงดันไฟฟ้าที่ Output หรือประสิทธิภาพของไมโครคอนโทรลเลอร์ ประมาณนี้

สมมุติว่าเราตัดสินใจ ซื้อบอร์ด Arduino มาใช้งาน โดยไม่รอฟังเสียงฟ้าเสียงฝน เราจะสามารถใช้งานบอร์ด Arduino ได้เลยทันทีที่หน้าจอคอมพิวเตอร์ แม้เรายังไม่มีโมดูลอิเล็กทรอนิกส์เลยสักชิ้นเดียว บอร์ด Arduino ก็มีโปรแกรมทดสอบ Examples Basic ให้เราสามารถทดลองใช้งานบอร์ด Arduino เบื้องต้นได้เลย และอาจจะยังไม่จำเป็นต้องพยายามทำความเข้าใจกับภาษา Arduino หรือภาษา C

 ซอฟต์แวร์ Arduino

ซอฟต์แวร์ Arduino ถูกสร้างขึ้นมาจากภาษา Arduino นั่นก็คือ ภาษา C/C++ นั่นเอง เช่นนั้นเราจะต้องใช้ภาษา C ในการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ โดยการสร้างและเขียนโปรแกรมชุดคำสั่ง "Sketch Code" เราจะต้องใช้ภาษา C ผ่านโปรแกรม Arduino IDE หรือ Arduino Web Editor ซึ่งเป็นเครื่องมือสำหรับเขียนโปรแกรมชุดคำสั่งในการควบคุมไมโครคอนโทรลเลอร์ให้เป็นไปตามที่เราต้องการ โดยทั้งนี้เครื่องมืออย่างโปรแกรม Arduino IDE และ Arduino Web Editor จะสามารถรวบรวมและยืนยันโปรแกรมชุดคำสั่ง Sketch Code ได้ ซึ่งหาก Sketch Code มีความผิดพลาด หรือเขียนภาษา C ไม่รู้เรื่อง "ประมาณว่าโปรแกรมไม่เข้าใจในสิ่งที่เราเขียนขึ้น" ระบบจะมีการแจ้งเตือนให้เราทำการแก้ไขในวรรคที่ผิด หรือบรรทัดที่ระบบไม่เข้าใจทันที และขั้นตอนสุดท้ายในการทำงานของซอฟต์แวร์ Arduino คือ การ Upload Sketch Code ไปยังบอร์ด Arduino หมายถึงการ Upload โปรแกรมชุดคำสั่ง (Sketch Code) ไปยังไมโครคอนโทรลเลอร์ของบอร์ด Arduino เพื่อที่จะให้บอร์ดส่ง-รับสัญญาณ Data หรือสัญญาณไฟ 3V หรือ 5V ไปยังวงจรโมดูลอิเล็กทรอนิกส์และควบคุมการทำงานให้เป็นไปตามที่เราต้องการ

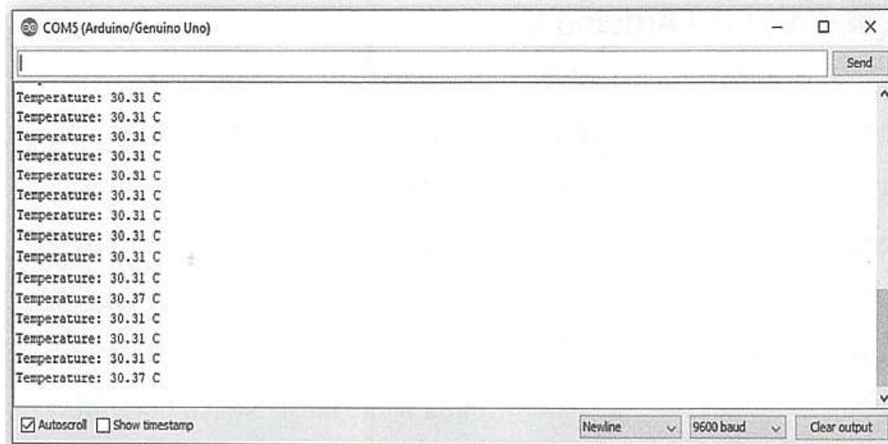
โปรแกรม Arduino IDE

โปรแกรม Arduino IDE คือ ซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์อย่างหนึ่ง ที่ถูกสร้างขึ้นมาเพื่อรองรับการใช้งานบอร์ด Arduino สำหรับการเขียนหรือสร้างโปรแกรมชุดคำสั่ง หรือ Sketch Code เรียกว่า Verify "การตรวจสอบหรือยืนยัน Sketch Code" หรือ Compile "การประมวลผล Sketch Code" และสำหรับขั้นตอนการ Upload Sketch Code ที่ผ่านการ Verify หรือ Compile ในสถานะ Success แล้ว ไปยังไมโครคอนโทรลเลอร์ของบอร์ด Arduino เพื่อให้เกิดการตอบสนองของชุดโมดูลอิเล็กทรอนิกส์ ตามโปรแกรม Sketch Code ได้กำหนดไว้ "ไฟติด ไฟดับ ติดกี่มิลลิวินาที ดับกี่มิลลิวินาที วนซ้ำภายในกี่วินาที"

นอกจากนี้โปรแกรม Arduino IDE ยังสามารถแสดงค่าการประมวลผลของไมโครคอนโทรลเลอร์ของบอร์ด Arduino ได้อีกด้วย โดยโปรแกรม Arduino IDE จะมีโหมด Serial Monitor สำหรับแสดงค่าเป็นตัวเลข และ Serial Plotter สำหรับแสดงค่าเป็นกราฟเส้น ซึ่งเราสามารถดูค่าตัวเลขและกราฟเส้นเหล่านี้ได้ที่เมนู Tools เช่น หากเรา



เขียน Sketch Code เพื่อให้บอร์ด Arduino รับสัญญาณจาก Temperature Humidity Sensor Module (เซ็นเซอร์ตรวจวัดค่าอุณหภูมิและความชื้น) บอร์ด Arduino จะทำการแปรสัญญาณ Data เป็นตัวเลขและแสดงใน Serial Monitor ซึ่งเราจะสามารถดูค่าอุณหภูมิและค่าความชื้นที่เซ็นเซอร์วัดได้แบบ Real Time หรือหากเราเขียน Sketch Code เพื่อ Code เพื่อให้บอร์ด Arduino รับสัญญาณจาก Heartbeat Sensor Module (เซ็นเซอร์ตรวจวัดสัญญาณชีพจร) บอร์ด Arduino จะทำการแปรสัญญาณ Data เป็นเส้นกราฟ ซึ่งเราจะสามารถดูค่าการเต้นของสัญญาณชีพได้แบบ Real Time เช่นกัน



สำหรับการเชื่อมต่อบอร์ด Arduino กับโปรแกรม Arduino IDE ก็คล้ายกันกับการเชื่อมต่อ Printer หรือเชื่อมต่อโทรศัพท์สมาร์ตโฟนกับเครื่องคอมพิวเตอร์ โดยการเชื่อมต่อบอร์ด Arduino จะเชื่อมต่อโดยสาย USB กับพอร์ตของเครื่องคอมพิวเตอร์ จากนั้นเข้าไปเลือกพอร์ตในโปรแกรม Arduino IDE โดยเฉพาะระบบปฏิบัติการคอมพิวเตอร์ที่มีประสิทธิภาพสูง เช่น Windows 10 และโปรแกรม USB ที่มีความเสถียร "เราไม่จำเป็นต้องค้นหาพอร์ตและเชื่อมต่อให้ยาก" เพราะระบบมันค้นหาและเลือกให้เองแบบอัตโนมัติ "ประมาณว่าเสียบ USB ปุ๊บ ก็เห็นปั๊บ ใช้ได้เลย... Enjoy"

 Arduino Web Editor

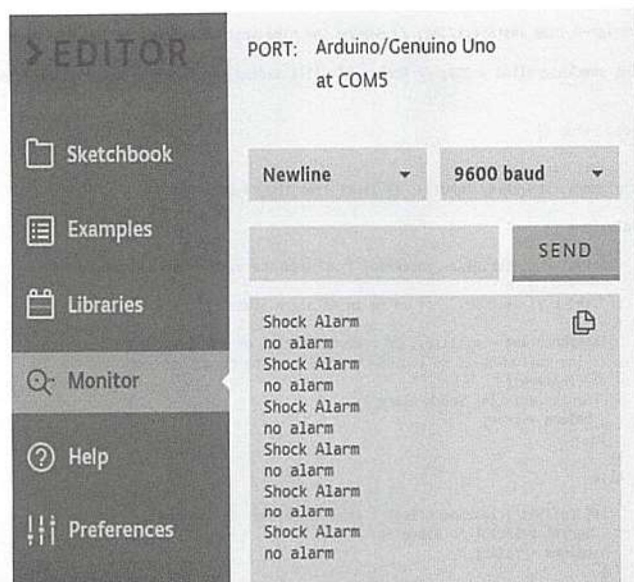
Arduino Web Editor คือ เว็บเพจหนึ่งของเว็บไซต์ www.arduino.cc โดยอยู่ที่ Link https://create.arduino.cc/ ซึ่งเว็บเพจนี้เปรียบเสมือนกับโปรแกรม Arduino IDE ทุกประการ

และใช้งานเหมือนกันทุกอย่าง Verify หรือ Compile และ Upload Sketch Code ไปยังบอร์ด Arduino ได้เช่นเดียวกับโปรแกรม Arduino IDE เพียงแต่ Arduino Web Editor เป็นเครื่องมือที่อำนวยความสะดวกให้การใช้ไมโครคอนโทรเลอร์ สามารถเกิดขึ้นได้ทุกที่ เราสามารถ Upload Sketch Code

ไปยังบอร์ด Arduino ได้ทุกที่ทุกเวลา โดยไม่ต้องติดตั้งโปรแกรม Arduino IDE บนระบบปฏิบัติการคอมพิวเตอร์เลยแต่อย่างใด

สมมุติว่าเราจำเป็นจะต้อง Upload Sketch Code ไปยังบอร์ด Arduino เพื่อควบคุมการทำงานของโมดูลเซ็นเซอร์ที่เราสร้างขึ้น เพื่อแสดงผลการทำงานของโมดูลให้เป็นที่ประจักษ์แก่สายตาของลูกค้า แต่โน้ตบุ๊คดันเจ๊งขึ้นมาเสียเฉยๆ "คุณจะแก้ปัญหานี้ยังไง" เราก็แค่ถือบอร์ด Arduino กับสาย USB ไปที่ร้านคอมฯ "ชั่วโมงละ 10 บาท" ต่อสาย USB จากบอร์ด Arduino เชื่อมต่อกับเครื่องคอมพิวเตอร์ในร้านอินเตอร์เน็ตคาเฟ่ จากนั้นเปิดเว็บไซต์ www.arduino.cc แล้วไปที่ LINK https://create.arduino.cc/ ลงทะเบียนเข้าใช้งาน เปิดไฟล์ Sketch Code ใหม่ พิมพ์ Sketch Code แล้วจัดการ Verify Sketch Code เสร็จแล้ว Upload Sketch Code ไปยังบอร์ด Arduino แค่นี้เป็นอันเสร็จพิธี

เมื่ออยู่ต่อหน้าลูกค้า เราก็แค่ต่อสาย Jumper Wire ต่อกับโมดูลเซ็นเซอร์ที่กำลังจะเป็นผลงานสร้างรายได้ของเรา แล้วเชื่อมต่อกับบอร์ด Arduino เสียบอะแดปเตอร์เข้าปลั๊กไฟ เพียงเท่านี้โมดูลเซ็นเซอร์ของเราก็ทำงานตามคุณสมบัติที่เราได้ตั้งค่ากำหนดไว้แล้ว



 Sketch Code

Sketch คือ แบบร่าง ถ้าเป็น Sketch Code ก็คือ แบบร่างรหัส ซึ่งเป็นภาษา Arduino หรือก็คือภาษา C/C++ ที่หลายคนกำลังเข็ดขยาดนั่นแหละ Sketch Code คือ โปรแกรมชุดคำสั่งที่ Upload ไปยังไมโครคอนโทรลเลอร์ของบอร์ด Arduino ให้สั่ง-รับสัญญาณไปยังโมดูลอิเล็กทรอนิกส์ ตามที่เราได้กำหนดในโปรแกรมชุดคำสั่ง อยากจะให้สวิตช์ติด ก็ใช้ภาษา C เขียนลงไปใน Sketch Code อยากให้ไฟ LED กะพริบ แล้วติดค้างกี่มิลลิวินาที ก็เขียนลงไปใน Sketch Code สำคัญที่สุด ภาษา C ที่เราใช้เขียนประกอบขึ้นเป็น Sketch Code บอร์ด Arduino จะต้องเข้าใจในโปรแกรมชุดคำสั่งของเรา ถ้าบอร์ดมันไม่เข้าใจภาษา C ของเรา มันก็ไม่ต่างอะไรกับก้อนหินก้อนหนึ่ง

Arduino ถือว่ากำลังได้รับความนิยม และกลายเป็นสิ่งที่น่าสนใจ โดยเฉพาะนักอิเล็กทรอนิกส์ทั้งมือใหม่และระดับเซียน สำหรับระดับเซียนทั้งหลาย พวกเขาจะเผยแพร่ประสบการณ์เกี่ยวกับ Arduino บนโลกอินเตอร์เน็ต ทำให้

sketch_jul19c.ino | ReadMe.adoc

```
int shockPin = 11; // Use Pin 10 as our Input
int shockVal = HIGH; // This is where we record our shock measurement
boolean bAlarm = false;

unsigned long lastShockTime; // Record the time that we measured a shock

int shockAlarmTime = 250; // Number of milli seconds to keep the shock alarm high

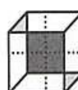

void setup ()
{
  Serial.begin(9600);
  pinMode (shockPin, INPUT) ; // input from the KY-002
}
void loop ()
{
  shockVal = digitalRead (shockPin) ; // read the value from our sensor

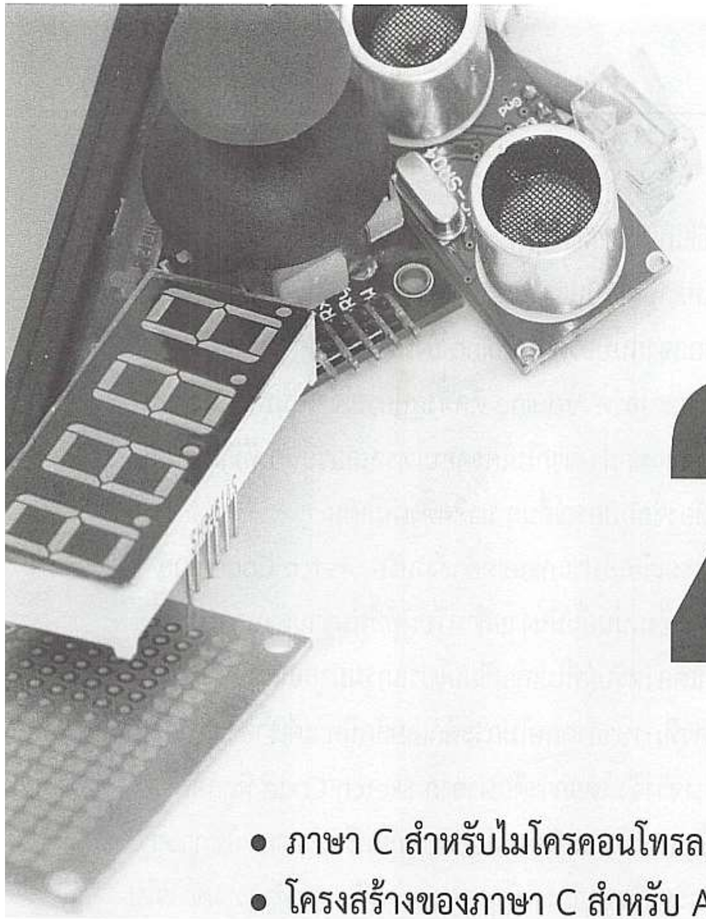
  if (shockVal == LOW) // If we're in an alarm state
  {
    lastShockTime = millis(); // record the time of the shock
    // The following is so you don't scroll on the output screen
    if (!bAlarm){
      Serial.println("Shock Alarm");
      bAlarm = true;
    }
  }
  else
  {
    if( (millis()-lastShockTime) > shockAlarmTime  &&  bAlarm){
      Serial.println("no alarm");
      bAlarm = false;
    }
```

เราสามารถหาเรียนรู้วิธีแก้ปัญหาต่างๆ ได้ง่าย หรือแม้แต่วิธีใช้งานหรือการสร้างโปรเจ็กต์มากมาย สิ่งเหล่านี้ถือเป็นแรงบันดาลใจที่สำคัญ ที่เกิดขึ้นกับนักเขียนโปรแกรมสมัครเล่น นอกจากนี้บอร์ด Arduino ยังเป็นไมโครคอนโทรลเลอร์ที่พร้อมใช้งานทันที เพราะบอร์ด Arduino ติดตั้งอุปกรณ์จำเป็นพื้นฐานมาให้เรียบร้อยทุกอย่างแล้ว ซึ่งแตกต่างจากไมโครคอนโทรลเลอร์อื่นๆ ที่มีแต่ตัวกับหัวใจเปล่าๆ จำเป็นที่จะต้องซื้ออุปกรณ์อื่นๆ มาติดตั้งเพิ่มเติม

Arduino สามารถเขียนโปรแกรมชุดคำสั่ง หรือ Sketch Code ด้วยไวยากรณ์ภาษา C/C++ ซึ่งแน่นอนมันง่ายสำหรับผู้ที่มีพื้นฐานด้านการเขียนโปรแกรมอยู่บ้างแล้ว แต่สำหรับผู้ที่ไม่เคยเขียนโปรแกรมมาก่อนเลย ถือเป็นเรื่องที่หนัก โดยพาะใครที่ภาษาอังกฤษไม่กระดิกเลยสักนิด แต่เราสามารถเริ่มต้นศึกษาภาษา C อย่างจริงจัง โดยการศึกษาจาก Sketch Code ที่ถูกต้องและสมบูรณ์ใช้งานได้ดี และนำไปเปรียบเทียบกับคู่มือหรือตำราการใช้ภาษา C แน่นอนหลักการศึกษาเรียนรู้แบบนี้ทำให้ทุกอย่างเกี่ยวกับการทำความข้าใจในภาษา C ดูจะง่ายขึ้น และจุดเด่นอีกข้อหนึ่งของการเป็น Open-Source สำหรับ Arduino คือ Libraries หาได้ไม่ยากบนอินเตอร์เน็ต ซึ่งมี Library ให้เลือกใช้มากมาย ทำให้การเขียนโปรแกรมทำได้ง่ายและรวดเร็วยิ่งขึ้น

แต่เหตุผลทุกอย่างที่กล่าวมา คงเทียบไม่ได้กับเหตุผลข้อนี้ ที่ทำให้บอร์ด Arduino ได้รับความนิยมสูงสุด คือ เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ราคาไม่แพงมากเกินไปสำหรับใครสักคนที่อยากจะเริ่มต้นสร้างประสบการณ์ใหม่ๆ กับ Arduino โดยเฉพาะเด็กที่กำลังต้องการพื้นฐานที่สำคัญเกี่ยวกับเทคโนโลยี บอร์ด Arduino จะช่วยเติมเต็มความคิดและความฝันให้พวกเขาได้อย่างไร้ซึ่งอุปสรรคที่เหมือนกำแพงขวางกั้นมันสมองของเด็กกับเทคโนโลยีที่ไม่เคยหยุดพัฒนา





2

- ภาษา C สำหรับไมโครคอนโทรลเลอร์
- โครงสร้างของภาษา C สำหรับ Arduino
- การประกาศคำสั่งในส่วน Header
- ฟังก์ชัน setup() และฟังก์ชัน loop()
- การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน
- ข้อกำหนดสำหรับการตั้งชื่อตัวแปร
- ชนิดและประเภทของตัวแปร

การเขียนภาษา C สำหรับไมโครคอนโทรลเลอร์

อย่างที่เคยบอกไปแล้วในบทแรก "อย่าหมกมุ่นกับภาษา C ให้มากนัก" เพราะถ้าเราลองเรียนรู้แล้ว "ไม่เข้าใจ" ยิ่งสับสนไม่เข้าใจภาษา C มากเท่าไหร่ ยิ่งเป็นการบั่นทอนกำลังใจตัวเองมากขึ้นเท่านั้น "สรุปยอมแพ้ตั้งแต่ยังไม่ทันได้สัมผัสบอร์ด Arduino" เอาเป็นว่าในบทนี้จะอธิบายถึงหลักการของการใช้ภาษา C เพื่อให้เข้าใจในเบื้องต้น ก่อนที่เราจะไปลงมือปฏิบัติลงมือจริง "เล่น Arduino ให้สนุก" ขอย้ำอีกครั้งว่า "อย่าซีเรียสกับบทนี้" เพราะยังมีวิธีทำให้เราเข้าใจการใช้ภาษา C อีกมากมาย เช่น หากเราได้ลองเชื่อมต่อโมดูลภายนอกกับบอร์ด Arduino และเชื่อมต่อกับระบบปฏิบัติการคอมพิวเตอร์แล้ว เราจะเข้าใจหลักการใช้ภาษา C มากยิ่งขึ้น แต่ถึงอย่างไรคงจะเป็นไปไม่ได้ "หรืออาจมีพื้นฐานไม่แน่นพอ" ถ้าจะลองเขียนโปรแกรมตาม YouTube เลยทันที โดยไม่มีพื้นฐานการเขียนโปรแกรมด้วยภาษา C เลยสักนิด

"เรามีคอมพิวเตอร์ มีบอร์ด Arduino มีชุดอุปกรณ์โมดูล มีบอร์ดทดลอง มีสาย Jumper" ดูเหมือนจะมีครบทุกอย่าง แต่ถ้าเราไม่มีชุดคำสั่งหรือโปรแกรมไว้สำหรับสั่งงานให้ไมโครคอนโทรลเลอร์ (Arduino) ทำงานตามที่ต้องการ ภาษาสายคอมฯ เขาเรียกว่า ซอฟต์แวร์ ก็ดูเหมือนว่าอุปกรณ์และเครื่องมือที่เรามีอยู่จะเปล่าประโยชน์ "เอาไปทำอะไรได้" เช่นนั้นเราจะต้องศึกษาเรียนรู้ถึงการเขียน

โปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์เสียก่อน "เพื่อพื้นฐานที่ดี" โดยการเขียนโปรแกรม คือ การสร้างชุดคำสั่งเพื่อโปรแกรมให้ไมโครคอนโทรลเลอร์ (Arduino) ทำงาน ซึ่งการทำงานของไมโครคอนโทรลเลอร์ คือ การสั่งการและควบคุมการทำงานของชุดโมดูล หรือชุดอุปกรณ์อิเล็กทรอนิกส์ที่ทำงานร่วมกันตามที่เราต้องการ

"ทำไมต้องภาษา C" เพราะภาษา C เป็นภาษาที่ใกล้เคียงกับภาษาต้นกำเนิดของระบบคอมพิวเตอร์ แน่นอนถ้าเราศึกษาเรียนรู้การใช้ภาษา C เราจะเข้าใจระบบคอมพิวเตอร์ได้มากยิ่งขึ้น โดยเฉพาะใครที่ใฝ่ฝันอยากจะเป็นโปรแกรมเมอร์ เพราะระบบปฏิบัติการมากมาย รวมถึงโปรแกรมขับต่างๆ "Drivers" ถูกเขียนขึ้นด้วยภาษา C นั่นแสดงว่า หากเราเขียนภาษา C เป็น เราก็สามารถสร้าง Drivers หรือระบบปฏิบัติการต่างๆ ได้เช่นกัน นอกจากนี้ภาษา C ถูกนำไปพัฒนาต่อยอดสำหรับโปรแกรมไมโครคอนโทรลเลอร์ เพราะภาษา C เป็นภาษาที่ระดับต่ำกว่าภาษาอื่นๆ (ง่ายกว่า) เช่น ภาษา C++ หรือ Java เป็นต้น แต่เป็นภาษาระบบปฏิบัติการคอมพิวเตอร์ขั้นพื้นฐาน ถ้าเราสามารถใช้ภาษา C ได้ เราก็จะสามารถศึกษาเรียนรู้ภาษาอื่นที่ยากกว่าได้อย่างรวดเร็ว "เข้าใจได้ง่ายขึ้น"

นอกจากนี้โปรแกรมที่เขียนด้วยภาษา C จะมีขนาดเล็กกว่า และเร็วกว่าโปรแกรมที่เขียนด้วยภาษาอื่นๆ ที่สำคัญที่สุด ภาษา C เป็น Open-Source เปิดกว้างให้ผู้ใช้งานสามารถพัฒนาต่อยอดได้ โดยระบบปฏิบัติการคอมพิวเตอร์ที่สร้างขึ้นมาจากภาษา C คือ ลีนุกซ์ (Linux) เป็นระบบปฏิบัติการที่ไม่มีลิขสิทธิ์ ใช้งานได้ฟรี "ใครจะเอาไปใช้ก็ได้ พัฒนาให้เป็นแบบอื่นก็ได้แต่การพัฒนาจะเกิดขึ้นบนพื้นฐานของระบบปฏิบัติการลีนุกซ์ นั่นแปลว่า ทั้งหมดจะเกิดขึ้นโดยใช้ภาษา C ในการพัฒนาและประยุกต์" อีกทั้งภาษา C มีมานานมาก โดยภาษา C เป็นภาษาโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ซึ่งเริ่มใช้กันอย่างแพร่หลายในปี พ.ศ. 2532 ทำให้มีโค้ดของภาษา C เกิดขึ้นมากมาย ประโยชน์เช่นนี้เองที่เราจะสามารถศึกษาอัลกอรึทึมใหม่ๆ หรือฟังชันจากโปรแกรมที่ถูกเขียนไว้แล้ว

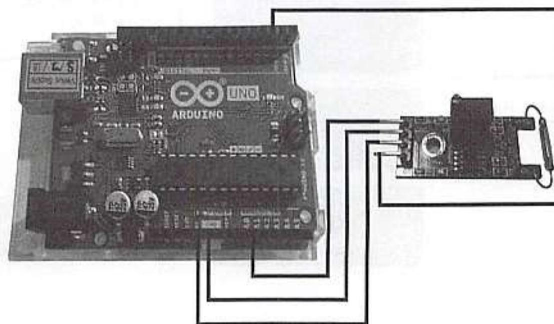
ด้วยความที่ภาษา C เป็นภาษาโปรแกรมพื้นฐานที่มีมานาน จึงทำให้หัวใจ

ของโมดูลอิเล็กทรอนิกส์มากมายถูกพัฒนาขึ้นด้วยภาษา C นั่นหมายถึง "ไมโครโปรเซสเซอร์" ทุกตัวรองรับภาษา C ไม่ว่าไมโครโปรเซสเซอร์รุ่นไหนปีใดก็ตาม "สามารถเขียนโปรแกรมด้วยภาษา C" อาจกล่าวได้ว่า ตั้งแต่ไม้ช็อตยุงไปจนถึงโทรศัพท์สมาร์ตโฟนที่เราใช้งาน "รองรับภาษา C" จึงเปรียบได้ว่าภาษา C คือพลังแห่งเทคโนโลยีของโลกเลยทีเดียว

 ภาษา C สำหรับไมโครคอนโทรลเลอร์

ไม่ว่าไมโครคอนโทรลเลอร์จะเป็นยี่ห้อใดรุ่นไหนหรือตระกูลอะไร จะสามารถทำงานได้ต่อเมื่อมีชุดคำสั่งที่สั่งให้ทำงานตามที่ต้องการ ชุดคำสั่งนี้ หมายถึง โปรแกรม ซึ่งชุดคำสั่งหรือโปรแกรมจะต้องใช้ภาษาที่ไมโครคอนโทรลเลอร์เข้าใจ และสามารถทำงานได้ตามที่โปรแกรมไว้ โดยจะทำงานอยู่ในรูปของรหัสลอจิก "0 และ 1" ซึ่งเป็นภาษาเครื่อง แน่นอนว่าภาษาเครื่องเป็นภาษาที่มนุษย์ไม่เข้าใจ "แล้วจะสร้างโปรแกรมอีท่าไหน" เช่นนั้นการใช้ภาษา C ที่อยู่ในรูปของภาษาเครื่อง "รหัสลอจิก" จึงมีความง่ายและรวดเร็วต่อการพัฒนาโปรแกรมใช้งานไมโครคอนโทรลเลอร์ เพราะมนุษย์ก็เข้าใจ เครื่องก็เข้าใจ เพียงแต่อ่านภาษาในคนละมุมมองเท่านั้น

ก่อนที่เราจะไปเริ่มขั้นตอนการปฏิบัติ "เล่น Arduino แบบสนุกสุดเหวี่ยง" แน่นอนเราจำเป็นจะต้องผ่านตากับการเขียนโปรแกรมด้วยภาษา C สำหรับบอร์ด Arduino เพราะถ้าเรา "ไม่ได้รู้อะไรบ้างเลย" เกี่ยวกับโครงสร้างการเขียนโปรแกรม เราจะเหมือนนั่งอ่านการ์ตูนญี่ปุ่นที่พิมพ์เป็นตัวอักษรฮิรานางะ



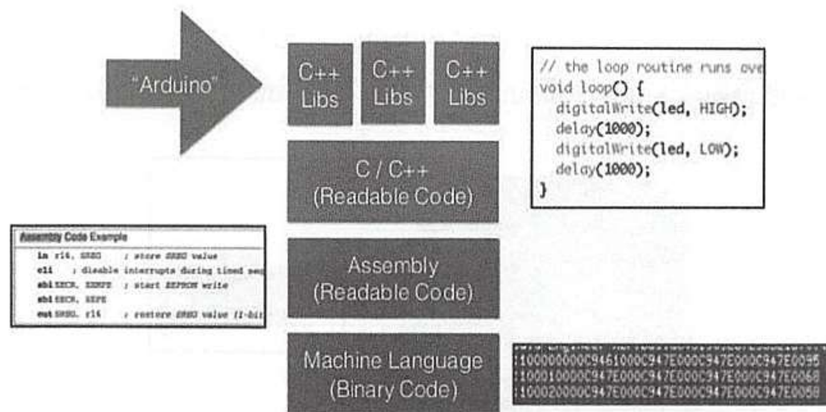
"ดูได้แต่ภาพ ไม่รู้ว่ามันพูดว่าอะไร" จงดูผ่านตาพอเข้าใจ หลักการเขียนโปรแกรม การกำหนดเงื่อนไข การเริ่มต้นและสิ้นสุด จากนั้นนำประสบการณ์การเรียนรู้เบื้องต้นเหล่านี้ ไปใช้หาคำตอบกับความสนุกสุดเหวี่ยงกับบอร์ด Arduino

โครงสร้างของภาษา C สำหรับ Arduino

ภาษา C สำหรับ Arduino ในการเขียนโปรแกรมจะจัดแบ่งรูปแบบโครงสร้างออกเป็นหลายส่วน ซึ่งแต่ละส่วนจะเรียกว่า ฟังก์ชัน แล้วจึงนำทุกฟังก์ชันมารวมเข้าด้วยกัน เรียกว่าโปรแกรม หรือโปรแกรมชุดคำสั่ง หรือ Sketch Code ทั้งนี้โครงสร้างการเขียน Sketch Code ของ Arduino ทุกโปรแกรม จะต้องประกอบไปด้วยฟังก์ชันหรือแต่ละส่วนจำนวนเท่าใดก็ได้ แต่ที่สำคัญที่สุด ทุกโปรแกรมชุดคำสั่ง จะต้องมีอย่างน้อย 2 ฟังก์ชัน คือ setup() และ loop()

ไม่ว่าเราจะมีประสบการณ์กับโครงสร้างของภาษา C สำหรับ Arduino มากหรือน้อยแค่ไหน เมื่อได้เห็น Sketch Code ผ่านตามาบ้างแล้ว เราจะสังเกตเห็นได้ว่า พื้นฐานโครงสร้างของ Sketch Code จะประกอบไปด้วย 3 ส่วนหลักๆ คือ

Arduino Langauge



- **Header**

Header คือ ส่วนหัวของโปรแกรม ซึ่งเป็นส่วนรวบรวมคำสั่งต่างๆ (Compiler Directive) และการประกาศตัวแปร รวมถึงค่าคงที่อื่นๆ เพื่อให้ระบบเข้าใจถึงตัวแปรและค่าเหล่านั้น อาจกล่าวได้ว่า ส่วนของ Header จะใช้เพื่อสรุปแนวทางคำสั่งเบื้องต้น หรือคำนำเพื่อให้ทราบถึงวัตถุประสงค์ของ Sketch Code นั้น ทั้งนี้ในส่วน Header จะมีหรือไม่ก็ได้ หมายถึง เราสามารถกำหนดหมุด Input หรือหมุด Output ในส่วนของ Header นี้ก็ได้ แต่ถ้า Sketch Code ของเราไม่มี Header ก็สามารถกำหนดหมุดการส่งและรับสัญญาณไว้ที่ส่วน void setup() ก็ได้ "Header ก็ไม่ต้องมี"

- **void setup()**

void setup() คือ ส่วนที่เป็นฟังก์ชันบังคับที่จะต้องกำหนดขึ้นในทุกโปรแกรม Sketch Code "ไม่มีไม่ได้" แม้เราจะมีการกำหนดหมุด Input หรือ Output ไว้ในส่วนของ Header แล้วก็ตาม "อาจไม่ต้องการใช้งานในส่วนนี้" แต่การเขียน Sketch Code จำเป็นจะต้องประกาศใน void setup() หรือเราอาจไม่ต้องเขียนอะไรในวงเล็บปีกกา { } ก็ได้ ทั้งนี้วงเล็บปีกกาจะเป็นสัญลักษณ์ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชัน หมายถึง การกำหนด การประกาศจะต้องเริ่มขึ้นหลังจากวงเล็บปีกกาเปิด { และปิดท้ายชุดคำสั่งด้วยวงเล็บปีกกาปิด } เสมอ สำหรับจุดประสงค์หลักที่การเขียน Sketch Code จะต้องมีส่วน void setup() ก็เพราะฟังก์ชันนี้จะใช้สำหรับใส่คำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียว ในช่วงเริ่มต้น Run โปรแกรม หลังจากที่การ Upload Sketch Code เสร็จสิ้น "Done Uploading" หรือ "Success" ประมาณว่า บอร์ด Arduino จะสั่งให้โมดูลอิเล็กทรอนิกส์ ทำงานครั้งแรกและครั้งเดียวสำหรับ Sketch Code ในส่วนของ void setup() นี้ และคำสั่งดังกล่าว จะต้องเป็นคำสั่งที่เกี่ยวกับการตั้งค่า Setup สำหรับค่าการทำงานต่างๆ หรือการกำหนดหมุดให้ส่ง-รับสัญญาณ PinMode และการกำหนดค่า Baud Rate หรืออัตราการส่งสัญญาณข้อมูล เช่น

จำนวนครั้งต่อวินาที การเปลี่ยนสถานะ ระดับ Voltage หรือความถี่ของสัญญาณ "ตั้งค่าให้ทำการแค่รอบเดียวจบ" ซึ่งการทำงานครั้งต่อไป เราจะต้องไปกำหนดคำสั่งและเงื่อนไขต่างๆ ในส่วนสุดท้าย คือ loop() เช่น ติดกี่มิลลิวินาที ดับกี่มิลลิวินาที ทำงานรอบเดียวจบ จนกว่า... หรือทำงานวนซ้ำไปเรื่อยๆ เป็นต้น

• void loop()

void loop() คือ ส่วนที่เป็นฟังก์ชันที่ระบบบังคับให้มีทุกโปรแกรมเช่นกัน ซึ่ง Sketch Code ส่วนนี้ จะใช้ใส่คำสั่งที่ต้องการให้โปรแกรมทำงานวนซ้ำไปเรื่อยๆ "ไม่มีจบสิ้น" โดยเมื่อบอร์ดส่งสัญญาณในส่วนของ void setup() สัญญาณคำสั่งนั้น จะส่งให้โมดูลอิเล็กทรอนิกส์จะทำงานเพียงรอบเดียว เมื่อมาถึงโปรแกรมในส่วนนี้ void loop() บอร์ด Arduino จะส่งสัญญาณคำสั่งให้โมดูลอิเล็กทรอนิกส์ทำงานวนซ้ำไปเรื่อยๆ ถ้าไม่มีคำสั่งอื่นที่ให้โปรแกรมออกจากวงวนนี้

```
#include <IRremote.h> //adds the library code to the sketch
const int irReceiverPin = 12; //receiver module S pin is connected to arduino D12
const int ledPin = 2;
IRrecv irrecv(irReceiverPin);
decode_results decodedSignal;
```

Header

```
void setup()
{
   Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    irrecv.enableIRIn();
}
boolean lightState = false;
unsigned long last = millis();
```

void setup()

```
void loop()
{
    if (irrecv.decode(&decodedSignal) == true) //this is true if a message has been received
    {
      Serial.println(decodedSignal.value, HEX);
        if (millis() - last > 250) {
            //has it been 1/4 sec since last message
            lightState = !lightState;
            //toggle the LED
            digitalWrite(ledPin, lightState);
        }
        last = millis();
        irrecv.resume();
        // watch out for another message
    }
}
```

void loop()

 สรุปโครงสร้างภาษา C สำหรับการเขียน Sketch Code

สำหรับใครที่ยังไม่เคยเรียนรู้การเขียนโปรแกรมในภาษาใดๆ มาก่อน โดยเฉพาะภาษา C จำเป็นจะต้องทำความเข้าใจเกี่ยวกับโครงสร้างที่สามารถนำไปใช้งานได้จริงดังนี้

1. **พรีโปรเซสเซอร์ไดเรกทีฟ** (Preprocessor directives) หรือส่วนของ Header
2. **ส่วนของการกำหนดค่า** (Global declarations) หรือการกำหนดค่าในส่วนของ Header
3. **ฟังก์ชัน setup() และฟังก์ชัน loop()**
4. **การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน** (Users-Defined Function)
5. **ส่วนอธิบายโปรแกรม** (Progarm Comments)

เช่นนั้นก่อนที่เราจะไปจับตัวบอร์ด Arduino เราควรจะมาศึกษาทำความเข้าใจกับโครงสร้างการเขียนโปรแกรมเบื้องต้นสำหรับ Arduino นี้ก่อน เพื่อให้เราเข้าใจใน Sketch Code ที่เรากำลังจะได้เห็นใน Project ต่างๆ ได้มากยิ่งขึ้น

1. พรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

สำหรับโครงสร้างแรกนี้โดยปกติเกือบทุกโปรแกรมจะต้องมี แต่ในความเป็นจริงจะมีหรือไม่มีก็ได้ โดยส่วนนี้จะเป็นส่วนที่ Compiler จะมีการประมวลผลและทำตามคำสั่งก่อนที่จะมีการ Verify เพื่อ Compile โปรแกรม Sketch Code ซึ่งจะเริ่มต้นด้วยเครื่องหมายไดเรกทีฟ บางคนเรียกสี่เหลี่ยม บางคนเรียกชาร์ป # จากนั้นตามด้วยชื่อคำสั่งที่ต้องการเรียกใช้ หรือต้องการกำหนด ทั้งนี้ส่วนนี้จะอยู่ในส่วนบนสุด หรือส่วนหัวของโปรแกรม (Header) และต้องอยู่นอกฟังก์ชันหลักอย่าง setup() และ loop()

#include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอก เพื่อเรียกใช้ฟังก์ชันหรือ

ตัวแปรที่มีการสร้างหรือกำหนดไว้ในไฟล์นั้น (ไฟล์ Libraries) โดยจะมีรูปแบบการใช้งาน คือ #include <ชื่อไฟล์.h> เช่น

```
#include <Wire.h>
#include <Time.h>
```

เราจะเห็นได้ว่ามีการอ้างอิงไฟล์ Wire.h และไฟล์ Time.h ซึ่งเป็น Libraries พื้นฐานที่มีอยู่ใน Arduino ซึ่งทำให้สามารถใช้ฟังก์ชันเกี่ยวกับเวลาที่ Libraries Time มีการสร้างไว้ให้ใช้งานได้ เช่น

```
#include "myFunction.h"
```

สำหรับตัวอย่างนี้ Compiler จะค้นหาไฟล์ myFunction.h ภายในโฟลเดอร์ของโปรแกรม Arduino IDE ทันที หากไม่พบก็จะแจ้งเป็นข้อผิดพลาดออกมา (Error)

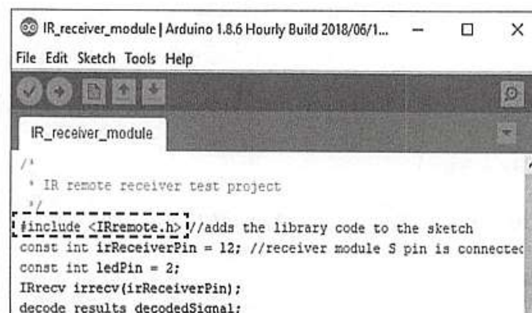
```
#define
```

คำสั่งนี้เป็นคำสั่งที่ใช้ในการแทนข้อความที่กำหนดไว้ ซึ่งจะไม่มีการอ้างอิงกับตัวโปรแกรมเลย เช่น

```
#define NAME VALUE
```

Header คำสั่ง Compiler Directive เพื่อเรียกใช้โปรแกรม

ทีนี้เรามาลงลึกถึงรายละเอียดการใช้คำสั่งในส่วนของพรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor Directives) หรือส่วนของ Header กันดีกว่า สำหรับการเขียน Header โดยขึ้นต้นด้วยเครื่องหมาย #... ซึ่งเป็นคำสั่งพิเศษที่เรียกว่า

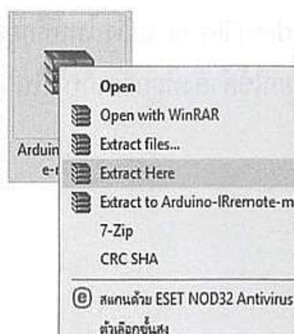


Compiler Directive เพราะเป็นเหมือนการประกาศมากกว่าคำสั่ง

เราจะสังเกตเห็นได้ว่า ทุกคำสั่งจะปิดท้ายประโยคด้วยเครื่องหมาย เซมิโคล่อน ; เสมอ จะมีเพียงคำสั่งพิเศษนี้คำสั่งเดียว #include <IRremote.h> ที่ไม่มีเครื่องหมาย ; ปิดท้ายประโยค เพราะคำสั่ง Compiler Directive จะใช้สำหรับบอกให้ Compiler หรือระบบทำหน้าที่รวบรวม รับรู้ถึงเงื่อนไขในการแปลคำสั่งเท่านั้น อย่างเช่นในกรณีคำสั่งนี้ #include <IRremote.h> สำหรับแจ้งให้ Compiler รับรู้ว่าในการแปลคำสั่งของ Sketch Code นี้ จะต้องมีไฟล์ภายนอกใดบ้างที่จำเป็นจะต้องใช้ร่วมในการแปลคำสั่งให้กับ Sketch Code นี้ โดยไฟล์ภายนอกนั้น หมายถึง Libraries ของ Sketch Code IRremote ที่เราดาวน์โหลดมาเพื่อให้ Sketch Code สามารถ Verify และ Upload ได้นั่นเอง (เราจะได้รับประสบการณ์มากมายสำหรับของการอัพเดต Libraries ที่โปรแกรม Arduino IDE หรือเว็บไซต์ Arduino Web Editor ไม่มี ในบทสุดท้าย Arduino Project)

ยกตัวอย่าง ถ้าเราจะเขียน Sketch Code เพื่อสร้างโปรแกรมชุดคำสั่งให้ IR Receiver Module ทำงาน (ตรวจจับแสงอินฟราเรด) เราจำเป็นจะต้องไปค้นหา Libraries ของ Sketch Code แล้วดาวน์โหลดไฟล์มาเก็บไว้ในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE หรือเว็บโปรแกรม Arduino Web Editor เมื่อทำการเก็บ Libraries ใหม่เข้าคลัง Libraries ของโปรแกรม เช่น Arduino IDE แล้ว





ดาวน์โหลด Libraries มาแล้ว ทำการแตกไฟล์ จะได้โฟลเดอร์ Libraries ที่ต้องการ จากนั้น Copy โฟลเดอร์ Libraries ไป Paste หรือวางในโฟลเดอร์ Libraries ของโปรแกรม เช่น Arduino IDE

Name	Date modified
drivers	6/18/2018 3:33 PM
examples	6/18
hardware	6/18
java	6/18
lib	6/18/2018 3:33 PM
libraries	7/14/2018 3:13 PM
reference	6/18/2018 3:33 PM

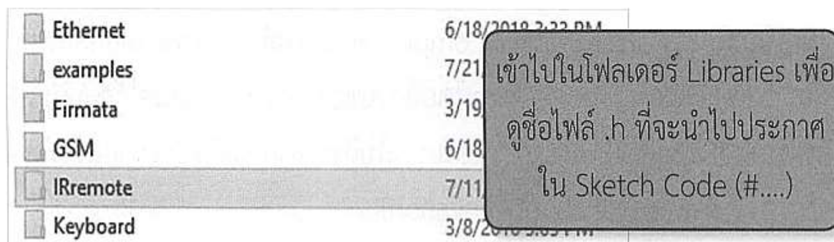
เข้าไปโฟลเดอร์ Libraries ของ
โปรแกรม เช่น Arduino IDE

Name	Date modified
Adafruit_Circuit_Playground	7/18/2017 11:2
Bridge	8/25/2016 5:20
DHT	7/14/2018 3:13
Esplora	5/20/2015 5:10
Ethernet	6/18/2018 3:33
Firmata	3/19/2017 12:4
GSM	6/18/2018 3:33
IRremote	7/11/2018 11:3
Keyboard	3/8/2016 5:05 PM
LiquidCrystal	8/10/2017 10:26 AM

ทำการ Paste หรือวางโฟลเดอร์
Libraries ของ IRremote ลง
ในโฟลเดอร์ Libraries เช่น
โปรแกรม Arduino IDE

หมายเหตุ : *เราสามารถเรียนรู้วิธีการ Upload Sketch Code ไปยังเว็บโปรแกรม Arduino Web Editor ที่บท การใช้งาน Arduino Online* "Arduino Web Editor เขียนโปรแกรมผ่านบราวเซอร์"

จากนั้นเราจะต้องเข้าไปดูชื่อไฟล์ Libraries เพื่อนำไปประกาศในบรรทัดคำสั่ง Compiler Directive (#....) ซึ่งโดยปกติแล้ว ไฟล์ Libraries จะเป็นไฟล์นามสกุล .h หรือเรียกอีกอย่างว่า Header File หมายถึง ไฟล์ที่อยู่ส่วนบนของประโยค main() ประกาศเอาไว้ เพื่อเรียกใช้คำสั่งที่ถูกจัดเก็บไว้ใน Header File นั้นๆ



นี่คือชื่อไฟล์ Libraries ที่เราจะต้องนำชื่อไป ประกาศในบรรทัดคำสั่ง Compiler Directive

Name	Date modified
examples	7/11/2018 11:31 PM
IRremote.cpp	5/5/2011 10:28 PM
IRremote.h	5/5/2011 10:28 PM
IRremoteInt.h	12/8/2014 10:17 PM
keywords	5/5/2011 10:28 PM
LICENSE	5/5/2011 10:28 PM
readme	5/5/2011 10:28 PM

Sketch Code ที่จะต้องเรียกใช้งานไฟล์ภายนอกอย่างไฟล์ Libraries เราจะต้องแจ้งให้ Compiler หรือระบบที่ทำหน้าที่รวบรวมคำสั่ง ทำการค้นหาไฟล์ภายนอกนั้น โดยแสดงชื่อไฟล์ไว้ในวงเล็บเครื่องหมายมากกว่าและน้อยกว่า (< และ >) เช่น #include <IRremote.h>

- #include คือ #ประกอบด้วย
- <IRremote.h> คือ ไฟล์ภายนอก (Libraries) ชื่อว่า IRremote.h

เมื่อคำสั่งพิเศษนี้ #include <IRremote.h> ปรากฏอยู่บรรทัดแรกของ Sketch Code ระบบ (Compiler) จะทำการค้นหาไฟล์ที่ชื่อ IRremote.h ในคลัง Libraries ของโปรแกรม เช่น Arduino IDE แล้วทำการเรียกใช้คำสั่งนั้นเพื่อให้บอร์ด Arduino ส่งคำสั่งไปยัง IR Receiver Module

2. ส่วนของการกำหนดค่า (Global Declarations) หรือการประกาศคำสั่งในส่วน Header

ในส่วนของคำสั่งในกลุ่ม Header นอกจากการเรียกใช้ไฟล์ภายนอก ซึ่งไม่ใช่คำสั่งโดยตรงที่ส่งผลให้โมดูลอิเล็กทรอนิกส์มีปฏิกิริยาตอบสนองต่อคำสั่ง

"มันเป็นเพียงสร้างความเข้าใจให้ Compiler ค้นหาไฟล์ Libraries ได้ถูกต้องเท่านั้น" ถ้าเราจะเขียน Sketch Code โดยมีส่วนของ Header กำหนดไว้ก่อนจะไปถึงส่วนของ void setup() เราอาจเขียนรวมไปถึงคำสั่งส่วนที่ใช้ประกาศสร้าง เช่น

- ตัวแปร (Variable Declaration)
- ค่าคงที่ (Constant Declaration)
- ฟังก์ชันต่างๆ (Function Declaration)

โดยส่วนนี้จะเป็นส่วนที่ใช้ในการกำหนดชนิดตัวแปรแบบนอกฟังก์ชัน หรือการประกาศฟังก์ชันเพื่อให้สามารถกำหนดหรือเรียกใช้ได้จากทุกส่วนของโปรแกรม เช่น

```
int pin = 13;
void blink(void) ;
```

ดังตัวอย่างคำสั่งการกำหนดค่าต่อไปนี้

```
#include <IRremote.h> //adds the library code to the sketch
const int irReceiverPin = 12; //receiver module S pin is connected to arduino D12
const int ledPin = 2;
IRrecv irrecv(irReceiverPin);
decode_results decodedSignal;
```

```
const int irReceiverPin = 12; //receiver module S pin is
connected to arduino D12
const int ledPin = 2;
```

คำสั่ง 2 บรรทัดนี้ คือ ส่วนที่ใช้ประกาศสร้างค่าคงที่ หรือ Constant Declaration คำว่า const ที่อยู่ต้นบรรทัด ย่อมาจากคำว่า Constant ส่วนคำต่อไป int ย่อมาจาก Integer หมายถึง ตัวแปรที่ใช้สำหรับการเก็บค่าตัวเลขที่เป็นจำนวนเต็ม ซึ่งจะมีค่าระหว่าง - 32768 ถึง 32767

หมายเหตุ

- Const คือ Constant Declaration หมายถึง ค่าคงที่

- Int คือ Integer หมายถึง ตัวแปรใช้สำหรับการเก็บค่าตัวเลขที่เป็นจำนวนเต็ม

บรรทัดที่ 2 หมายถึง

- const int irReceiverPin = 12; หมายถึง ค่าคงที่ซึ่งเป็นตัวแปรที่ใช้สำหรับการเก็บค่าตัวเลข คือ หมุด 12
- //receiver module S pin is connected to arduino D12 หมายถึง กำหนดให้ขา S ของโมดูลเชื่อมต่อกับหมุด D12 ของบอร์ด Arduino

บรรทัดที่ 3 หมายถึง

- const int ledPin = 2 หมายถึง เมื่อได้สัญญาณค่าคงที่ซึ่งเป็นตัวแปรที่ใช้สำหรับการเก็บค่าตัวเลขแล้ว ให้ไฟ LED ติด โดยส่งสัญญาณผ่านหมุด 2

และบรรทัดที่ 4 และ 5 หมายถึง

```
4 IRrecv irrecv(irReceiverPin);
5 decode_results decodedSignal;
```

- IRrecv irrecv(irReceiverPin); หมายถึง IR receiver IR receiver (หมุด IR receiver) "คือ หมุด 12 อ้างอิงคำสั่งบรรทัดที่ 2"
- decode_results decodedSignal; หมายถึง ผลการถอดรหัสให้เป็นเป็นสัญญาณดิจิตอล

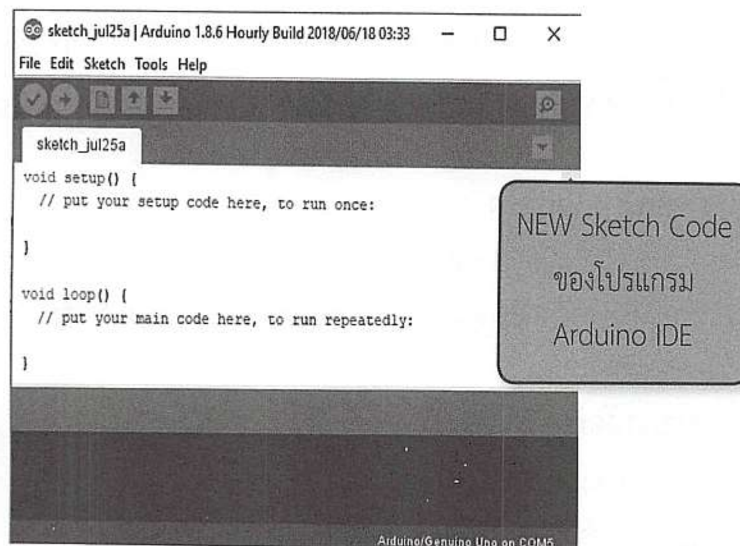
นอกจากนี้เรายังสามารถกำหนดค่าเพื่อไม่ให้ตัวแปรเปลืองพื้นที่หน่วยความจำ (RAM) ซึ่งจะส่งผลให้ CPU ไม่ต้องทำงานหนัก เช่น

```
1 #define LEDPIN 13
```

เช่นนั้นไม่ว่า LEDPIN จะอยู่ส่วนใดของ Sketch Code ก็ตาม Compiler จะแทนคำว่า LEDPIN ด้วยเลข 13 ซึ่งทำให้เราไม่ต้องสร้างเป็นตัวแปรขึ้นมาเพื่อเปลืองพื้นที่หน่วยความจำ อีกทั้งยังช่วยให้โปรแกรมทำงานได้เร็วยิ่งขึ้น เนื่องจาก CPU ไม่จำเป็นต้องขอข้อมูลมาจากหน่วยความจำหลายต่อนั่นเอง

3. ฟังก์ชัน setup() และฟังก์ชัน loop()

สำหรับคำสั่งทั้ง 2 ส่วนนี้ มีความสำคัญมากที่สุด ชนิดที่ว่าไม่เขียนไม่ได้ เช่น ในขั้นตอนของการสร้างไฟล์ Sketch Code ใหม่ ไม่ว่าจะเป็นโปรแกรม Arduino IDE หรือเว็บโปรแกรม Arduino Web Editor ระบบจะแสดงคำสั่งพื้นฐาน 2 ส่วนนี้ใน Sketch Code ใหม่โดยอัตโนมัติ



ระบบจะแสดงคำสั่ง 2 ส่วนที่สำคัญขึ้นมาในไฟล์ Sketch Code โดยอัตโนมัติ เราจะเห็นข้อความอธิบายที่ระบบแจ้งให้เราทราบถึงขั้นตอนการเขียน Sketch Code โดยฟังก์ชัน setup() และฟังก์ชัน loop() จะเป็นคำสั่งที่ถูกบังคับให้ต้องมีในทุกโปรแกรม โดยฟังก์ชัน setup() จะเป็นฟังก์ชันแรกที่ถูกเรียกใช้ในการกำหนดค่า หรือเริ่มต้นใช้งานของ Libraries ต่างๆ เช่น ในฟังก์ชัน setup() จะมีคำสั่ง pinMode() เพื่อกำหนดให้หมุดใดๆ ก็ตามที่เป็นดิจิตอลอินพุตหรือเอาต์พุต ส่วนฟังก์ชัน loop() จะเป็นฟังก์ชันที่ทำงานหลังจากฟังก์ชัน setup() เมื่อ setup() ได้ทำงานเสร็จสิ้นไปแล้ว จากนั้นจะวนเริ่มต้นใหม่แบบไม่รู้จบใน

ฟังก์ชัน loop() โดนการทำงานครบตามคำสั่งแล้ว ฟังก์ชัน loop() ก็จะถูกเรียกขึ้นมาใช้อีกไม่จบสิ้น

void setup() {

// put your setup code here, to run once: *หมายถึง เขียน Code ติดตั้งของคุณไว้ที่นี่เพื่อใช้งานครั้งเดียว*

void loop() {

// put your main code here, to run repeatedly: *หมายถึง เขียน Code หลักของคุณไว้ที่นี่เพื่อเรียกใช้ซ้ำ*

เช่นนั้นเราสามารถลบบรรทัดคำอธิบายเหล่านี้ออก แล้วแทนที่ด้วย Sketch Code ของเราได้เลย (ลบเฉพาะคำอธิบาย อย่าลบ void setup() หรือ void loop() และเครื่องหมาย { })



สำหรับไฟล์ Sketch Code ใหม่ของเว็บ Arduino Web Editor สามารถเขียน Sketch Code หลังเครื่องหมายปีกกาเปิด { ได้เลย โดยให้เครื่องหมายปีกกาปิด } อยู่ในบรรทัดถัดไป

ตัวอย่างการเขียนคำสั่ง void setup()

```
void setup(){
 Serial.begin(9600);
 pinMode(ledPin, OUTPUT);
 irrecv.enableIRIn();
}
```

เริ่มฟังก์ชัน

คำสั่งติดตั้งโปรแกรมใน void setup() จะเริ่มเขียนหลังเครื่องหมายปีกกาเปิด { Serial... และจบด้วยเครื่องหมายปีกกาปิดในบรรทัดต่อไป

จบฟังก์ชัน

ตัวอย่างการเขียนคำสั่ง void loop()

```
void loop(){
 if (irrecv.decode(&decodedSignal) == true) //this is true if a message has been received
 {
  Serial.println(decodedSignal.value, HEX);
   if (millis() - last > 250) {
    //has it been 1/4 sec since last message
    lightState = !lightState;
    //toggle the LED
    digitalWrite(ledPin, lightState);
   }
```

เริ่มฟังก์ชัน

จบฟังก์ชันและวนซ้ำ

คำสั่ง void loop() นี้ มีคำสั่งย่อยถึง 3 ข้อ จะสังเกตได้ว่า ทุกคำสั่งย่อยที่เขียนขึ้น จะเริ่มต้นหลังเครื่องหมายปีกกาเปิด { คำสั่งที่ 1 if...{ คำสั่งที่ 2 Serial... { คำสั่งที่ 3 //has... และจบด้วยเครื่องหมายปีกกาปิด } ในบรรทัดต่อไป

เราจะเห็นได้ว่าฟังก์ชันทั้ง 2 ส่วนนี้มีรูปแบบโครงสร้างที่คล้ายกัน แต่ถูกกำหนดด้วยชื่อและฟังก์ชันโดยเฉพาะ คือ void setup() และ void loop() โดยฟังก์ชันส่วนของ void setup() จะถูกเขียนกำหนดขึ้นก่อน void loop() ซึ่งจะถูกเขียนกำหนดขึ้นในขอบเขตของเครื่องหมายปีกกาเปิดและปิด {......} ตามตัวอย่าง Sketch Code ในส่วนของ void setup() และ void loop() ข้างต้น

หน้าที่หลักของฟังก์ชัน void setup() และ void loop()

เมื่อเรียนรู้วิธีการเขียน void setup() และ void loop() ไปแล้ว ทีนี้เรามาดูหน้าที่หลักของทั้ง 2 ฟังก์ชันนี้กัน เริ่มต้นด้วยหน้าที่ของฟังก์ชัน void setup() คือ เป็นส่วนของโปรแกรมย่อย สำหรับใช้บรรจุคำสั่งต่างๆ เพื่อกำหนดการทำงานของระบบ และรวมถึงเพื่อกำหนดคุณสมบัติการทำงานให้กับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ โดยคำสั่งทั้งหมดในส่วนของ void setup() จะถูกเรียกขึ้นมาใช้เพียงรอบเดียว ในช่วงเริ่มต้นของการทำงานตามโปรแกรม ไม่ว่าจะเป็นหลังจากการ Upload Sketch Code ไปยังบอร์ด หรือหลังจากกดปุ่ม Reset เพื่อให้โปรแกรมเริ่มต้นใหม่ ซึ่งปกติคำสั่งในส่วนนี้มักจะใช้เป็นคำสั่งสำหรับการกำหนดโหมดการทำงานของหมุดดิจิตอล หรือคำสั่งเพื่อกำหนดคุณสมบัติของพอร์ตสื่อสารแบบอนุกรม

```
void setup()
{
   คำสั่งต่างๆ ที่กำหนดขึ้นภายใต้คำสั่งในส่วนของ void setup()
}
```

ส่วนหน้าที่ของฟังก์ชัน void loop() คือ เป็นส่วนของโปรแกรมหลัก เพื่อใช้บรรจุคำสั่งควบคุมการทำงานต่างๆ ของโปรแกรมที่กำหนดให้โปรแกรมทำงาน

โดยคำสั่งในส่วนของฟังก์ชันนี้ จะถูกเรียกมาใช้ในการทำงานแบบซ้ำ "วนซ้ำไปเรื่อยๆ" ตั้งแต่เงื่อนไขแรกจนถึงเงื่อนไขสุดท้ายตามลำดับ แล้ววนซ้ำเริ่มต้นทำงานแบบนี้ไปเรื่อยๆ

```
void loop()
{
   คำสั่งต่างๆ ที่กำหนดขึ้นภายใต้คำสั่งในส่วนของ void loop()
}
```

ตัวอย่างคำสั่ง void setup() และ void loop()

```
int pin = 13;
void setup() {
 pinMode(pin, OUTPUT);
}
void loop() {
 digitalWrite(pin, HIGH);
 delay(1000);
 digitalWrite(pin, LOW);
 delay(1000);
        }
```

ตามตัวอย่าง Sketch Code ด้านบน เราจะสังเกตเห็นได้ว่ามีการประกาศตัวแปรแบบนอกฟังก์ชัน ทำให้สามารถกำหนดหรือเรียกใช้จากในฟังก์ชันใดก็ตาม

ได้ สำหรับฟังก์ชัน setup() ได้มีการกำหนดให้หมุดที่ 13 เป็นดิจิตอลเอาต์พุต ส่วนในฟังก์ชัน loop() ได้มีการกำหนดให้หมุด 13 มีค่าลอจิกเป็น 1 โดยจะใช้ฟังก์ชัน delay() ในการหน่วงเวลา 1 วินาที แล้วจึงกำหนดให้หมุด 13 มีค่าลอจิกเป็น 0 แล้วจึงหน่วงเวลา 1 วินาที จบฟังก์ชัน loop() และจะเริ่มทำฟังก์ชัน loop() ใหม่ ผลที่ได้คือไฟกะพริบบนบอร์ด Arduino Uno ในหมุดที่ 13 ทำงานวนไปแบบไม่มีสิ้นสุด และนี่ก็คือ Sketch Code ของ Blink ซึ่งเป็นโปรแกรม Example Basic ของโปรแกรม Arduino IDE นั่นเอง ซึ่งเราจะได้ทดลอง Project นี้เป็นโปรแกรมแรก หลังจากที่ได้ทำการติดตั้งโปรแกรม Arduino IDE เสร็จเรียบร้อยแล้ว (ในบทการดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE)

4. การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-Defined Function)

สำหรับการสร้างฟังก์ชัน คำสั่งทั้งหมดที่อยู่ภายในฟังก์ชัน ต้องอยู่ภายในขอบเขตของเครื่องหมายปีกกาเปิด { และปีกกาปิด } เท่านั้น โดยภายในขอบเขตของเครื่องหมายปีกกานี้ เราจะสามารถนำฟังก์ชันหรือคำสั่งใดๆ ก็ตามได้มาใส่ไว้ แต่จะต้องคั่นแต่ละคำสั่งด้วยเครื่องหมายเซมิโคล่อน ; ซึ่งเราจะนำคำสั่งต่างๆ บรรจุไว้ในบรรทัดเดียวกันเลยก็ได้ "แค่คั่นด้วย ;" หรืออาจจะแยกบรรทัดกันก็ได้ เพื่อความสวยงามของโค้ด และแน่นอนมันไม่มีผลกับขนาดของโปรแกรมหลังจากที่เราคลิก Verify และระบบทำการ Compile เช่น

```
void Mode(int pin) {
 pinMode(pin, OUTPUT);
}
void setup() {
 Mode(13);
}
```

5. ส่วนอธิบายโปรแกรม (Progarm Comments)

ส่วนของการอธิบายโปรแกรม หรือถ้าจะให้เข้าใจง่ายๆ ก็คือ การแสดงคำอธิบายของโปรแกรม ซึ่งถือเป็นส่วนที่สำคัญอย่างมากที่จะช่วยให้ผู้ที่ไม่ได้เขียนโปรแกรม "Copy Sketch Code เขามา" หรือเป็นผู้เขียนโปรแกรมเอง ให้สามารถเข้าใจโปรแกรมได้ง่ายยิ่งขึ้น โดยการอ่านจากคำอธิบายเพื่อการทำความเข้าใจโปรแกรม ซึ่งจะต้องอ่านแต่ละฟังก์ชัน โดยส่วนของการอธิบายโปรแกรมนี้ จะไม่ส่งผลกับขนาดของโปรแกรมหลัง Compile เพราะส่วนนี้จะถูกตัดทิ้งทั้งหมดเนื่องจากไม่ได้นำไปใช้งาน แต่จะมีผลเพียงเล็กน้อยก็คือ ไฟล์ Sketch Code จะมีขนาดใหญ่นิดหน่อย สำหรับการแสดงคำอธิบาย Code เยอะ แต่ถึงอย่างไรขนาดไฟล์ก็จะเพิ่มขึ้นตามตัวอักษรที่เราเขียนเพิ่มขึ้น เช่นนั้นการแสดงคำอธิบาย Code จึงไม่คิดพื้นที่มากจนส่งผล แต่ขอแนะนำให้แสดงคำอธิบาย Code ในรูปแบบสั้นและกระชับ ทั้งนี้เพื่อให้เกิดความรวดเร็วในการทำความเข้าใจ และไม่ยาวเกินไปจนต้องขยับหน้าต่างไปด้านขวา

สำหรับการแสดงคำอธิบาย Code จะมีอยู่ 2 รูปแบบด้วยกัน คือ เปิดด้วย /* และปิดด้วย */ โดยการแสดงคำอธิบาย Code แบบข้ามบรรทัด หมายถึง หากยังไม่มี */ เพื่อปิดการแสดงคำอธิบายในส่วนนั้น จะถือเป็นคำอธิบายทั้งหมดดังต่อไปนี้

ตัวอย่างแบบที่ 1

```
/*
This code by IOXhop.com
17/5/2558
*/
void setup() { .... }
```

ตัวอย่างแบบที่ 2

เป็นการแสดงคำอธิบายแบบบรรทัดเดียว คือ เปิดด้วยเครื่องหมาย // และปิดด้วยการขึ้นบรรทัดใหม่

```
void setup() {
 pinMode(13, OUTPUT); // Set pin 13 to output
```

 ตัวแปรของภาษา C สำหรับ Arduino

ตัวแปร หมายถึง ตัวอักษร หรือตัวเลข รวมถึงเครื่องหมายมากมายที่รวมกันเป็นชื่อ เพื่อใช้กำหนดเป็นตัวแทนของค่าข้อมูลที่ต้องการจะอ้างถึงใน Sketch Code หรือโปรแกรมการทำงาน เพราะสำหรับการทำงานของโปรแกรมนั้น จะใช้ค่าตัวเลขที่กำหนดขึ้นมานี้มาทำการประมวลผล เช่นนั้นการเขียนโปรแกรมจึงจะต้องเขียนให้มีการกำหนดเป็นค่าตัวเลข ถ้าจะเขียนตัวเลขใน Sketch Code ที่เราสร้างขึ้นแบบตรงๆ ไม่มีตัวแทน ย่อมทำให้โปรแกรมที่เกิดขึ้นเต็มไปด้วยค่าตัวเลขมากมาย แน่นอนนอกจากจะยากต่อการอ่านแล้ว ทำความเข้าใจยาก และยากต่อการตรวจสอบความถูกต้องอีกด้วย ดังนั้นจึงมีการกำหนดชื่อขึ้นมาใช้งานแทนค่าตัวเลข เพื่อให้การเขียน Sketch Code สะดวก หรือง่ายต่อการอ่านและทำความเข้าใจ ทั้งนี้ตัวแปรอาจมีทั้งแบบที่เป็นค่าที่สามารถเปลี่ยนแปลงได้ หรือ Variable และเป็นค่าแบบคงที่ ซึ่งไม่สามารถเปลี่ยนแปลงได้ หรือ Constant โดยการประกาศใช้งานตัวแปร จะต้องประกาศชนิดของตัวแปร หรืออาจกำหนดค่าเริ่มต้นให้กับตัวแปรด้วยก็ได้

ถ้าจะเปรียบเทียบว่า ตัวแปรนั้น เหมือนกับอะไร ก็อาจจะยกตัวอย่างง่ายๆ ให้เข้าใจได้ไม่ยาก คือ สมมุติว่า ถ้าจะต้องส่งเพื่อนไปโรงเรียน เพื่อน 1 คน เราจะใช้พาหนะใดไปส่งถึงจะรวดเร็วและคุ้มค่ามากที่สุด แน่นอนเราน่าจะใช้จักรยานยนต์ แต่ถ้าเพื่อน 10 คน เราคงใช้จักรยานยนต์ไม่ได้ ต้องใช้รถยนต์ถึงจะ

สามารถไปส่งเพื่อนทั้งสิบคนที่โรงเรียนได้ เช่นนั้นตัวแปรก็เปรียบได้กับรถขนของ ซึ่งมีขนาดแตกต่างกัน เพื่อให้สะดวกเหมาะสมและเกิดการคุ้มค่ามากที่สุด ซึ่งหลายคนอาจจะเคยเห็นตัวแปรที่ใช้บ่อยใน Sketch Code เช่น

- **int** จะใช้เก็บตัวเลขจำนวนเต็มไม่มีทศนิยม โดยจะใช้พื้นที่ 2 byte เก็บเลขได้ค่า 32,767 ถึง -32,768
- **double** จะใช้เก็บเลขทศนิยม โดยจะใช้พื้นที่ 4 byte เก็บเลขได้ค่า 3.4028235E+38 ถึง -3.4028235E+38

เพราะเหตุใดเราถึงจะต้องใช้ตัวแปรให้มีเหมาะสม และให้เกิดความคุ้มค่ามากที่สุด เพราะตัวแปรแต่ละตัวจะใช้พื้นที่และหน่วยความจำไม่เท่ากัน ถ้าใช้ตัวแปร double ไปเก็บเลข 5000 เราจะต้องใช้พื้นที่ถึง 4 byte ในการเก็บ แต่หากเราเลือกใช้ตัวแปร int ไปเก็บเลข 5000 จะใช้พื้นที่เพียง 2 byte ซึ่งใช้พื้นที่น้อย และยังส่งผลถึงความเร็วในการทำงานให้โปรแกรมได้ด้วย ถ้าจะต้องไปส่งเพื่อนคนเดียว แต่เอารถยนต์ไปส่ง รถยนต์คันใหญ่เดินทางไม่สะดวก ทั้งรถติดแถมไม่มีที่จอด แต่ถ้าเอาจักรยานยนต์ไปส่ง คันเล็กแล่นฉิวปราดเปรียวซอกแซกได้ถึงที่หมายอย่างรวดเร็ว

 ข้อกำหนดสำหรับการตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรสำหรับการเขียน Sketch Code จะต้องยึดหลักของ ANSI-C ทุกประการ ANSI-C หมายถึง ข้อกำหนดมาตรฐานสำหรับการเขียนโปรแกรมด้วยภาษา C นั่นเอง

1. ชื่อของตัวแปรจะต้องประกอบไปด้วยตัวอักษร ตัวเลข และจะต้องไม่มีเครื่องหมายอื่นปะปน แต่จะยอมให้ใช้เครื่องหมายพิเศษเพียง 2 ตัวเท่านั้น คือ Under Line (_) และ Dollar Sign ($) โดยเครื่องหมาย 2 ตัวนี้ สามารถนำมาใช้ในการตั้งชื่อได้

2. ชื่อของตัวแปรจะต้องเรียงติดทั้งหมด ห้ามเว้นวรรค และจะต้องมีความยาวไม่เกิน 32 ตัวอักษร
3. ชื่อของตัวแปร ไม่ว่าจะเป็นตัวอักษรทั้งหมด หรือตัวอักษรผสมกับตัวเลข และอาจจะผสมกับเครื่องหมายพิเศษ 2 ตัว ตามข้อ 1 ด้วยก็ได้ แต่สำคัญก็คือ จะต้องเริ่มต้นชื่อด้วยตัวอักษรเป็นลำดับแรกเสมอ
4. ชื่อของตัวแปรห้ามเริ่มต้นด้วยตัวเลข หรือเครื่องหมาย
5. สำหรับภาษา C ตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ ถือว่าเป็นคนละชื่อกัน เช่น LED Led และ led ทั้ง 3 ชื่อนี้ แม้จะมีความหมายเดียวกัน แต่ภาษา C ถือว่าเป็นคนละชื่อ และมีความหมายที่แตกต่างกันอีกด้วย
6. ชื่อตัวแปรห้ามนำคำสงวนมาใช้ตั้งชื่อ ซึ่งคำสงวนสำหรับการเขียน Sketch Code ของ Arduino หมายถึง ชื่อคำสั่ง และชื่อ Internal Function ที่สร้างไว้แล้วในภาษา C อยู่แล้ว ดังนี้

- คำสงวนที่เป็น Constant ของ Arduino

HIGH	LOW	INPUT	OUTPUT
SERIAL	DISPLAY	PI	HALF_PI
TWO_PI	LSBFIRST	MSBFIRST	CHANGE
FALLING	RISING	false	truenull

- คำสงวนที่เป็น Port Variables & Constants ของ Arduino

DDRB	PINB	PORTB	PB0	PB1
PB2	PB3	PB4	PB5	PB6
PB7	DDRC	PINC	PORTC	PC0
PC1	PC2	PC3	PC4	PC5
PC6	PC7	DDRD	PIND	PORTD
PD0	PD1	PD2	PD3	PD4
PD5	PD6	PD7		

- คำสงวนที่เป็น Datatypes ของ Arduino

boolean	byte	char	class	default
do	double	int	long	private
protected	public	return	short	signed
static	switch	throw	try	unsigned
void				

- คำสงวนที่เป็นคำสั่งของ Arduino

abs	acos	+=
+	[]	asin
=	atan	atan2
&	|	boolean
byte	case	ceil
char	class	,
//	?:	constrain
cos	{}	--
default	delay	delayMicroseconds
/	/**	.
else	==	exp
false	float	floor
for	<	<=
if	++	!=
int	<<	<
<=	log	&&
!	||	loop
max	millis	min
-	%	/*

*	new	null
()	PI	return
>>	;	Serial
Setup	sin	sq
sqrt	-=	switch
tan	this	true
void	while	begin
read	write	print
println	available	digitalWrite
digitalRead	analogRead	analogWrite
attachInterrupts	detachInterrupts	beginSerial
serialWrite	serialRead	serialAvailable
printString	printInteger	printByte
printHex	printOctal	printBinary
printNewline	pulseIn	shiftOut

 ชนิดและประเภทของตัวแปร

ในภาษา C จะมีการกำหนดและจำแนกชนิดของตัวแปรไว้ 5 ชนิด ซึ่งแต่ละชนิดจะมีคุณสมบัติการใช้งานที่ต่างกัน เพื่อใช้ในการเก็บข้อมูลที่มีรูปแบบแตกต่างกัน คือ

1. char ใช้เก็บข้อมูลที่เป็นตัวอักษร (Character) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 256 ค่า
2. int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (Integer) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 65536 ค่า
3. float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision

4. double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่งสามารถเก็บค่าตัวเลขทศนิยมที่มีความละเอียดและถูกต้องของทศนิยมมากกว่าแบบ float ถึง 2 เท่า
5. void ใช้เก็บตัวแปรที่ไม่มีค่า

ชนิดตัวแปร	จำนวนบิต	ค่าข้อมูลที่เก็บได้
char	8	-128 ถึง +127
int	16	-32768 ถึง +32767
float	32	3.4E ถึง 3.4E+38
double	64	1.7E-308 ถึง 1.7E+308
void	0	ไม่มีค่า

 คุณสมบัติเฉพาะของตัวแปร

เราคงเห็น 2 ตัวแปรนี้ ใน Sketch Code กันจนชินตา “char และ int” ซึ่งเป็นตัวแปรชนิดที่ใช้เก็บค่าตัวเลขที่เป็นจำนวนเต็ม แต่ภาษา C ไม่ได้มีการจำแนกว่า ตัวแปรไหนจะใช้เพื่อเก็บค่าตัวเลขที่เป็นค่าบวก หรือค่าลบ เช่นนั้นภาษา C จะใช้วิธีเพิ่มคำสั่งเพื่อกำหนดคุณสมบัติเฉพาะให้กับตัวแปรไว้อีก 4 คำสั่ง คือ

- unsigned ใช้เพื่อเก็บค่าตัวเลขจำนวนเต็มในตัวแปรเฉพาะค่าตัวเลขที่เป็นบวก (+) เท่านั้น
- signed ใช้เพื่อเก็บค่าตัวเลขจำนวนเต็มในตัวแปรทั้งค่าบวก (+) และค่าลบ (-)
- short ใช้เพื่อเก็บค่าตัวเลขจำนวนเต็มในตัวแปรที่มีค่าน้อยกว่า int
- long ใช้เพื่อเก็บค่าตัวเลขจำนวนเต็มในตัวแปรที่มีค่ามากกว่า int เป็น 2 เท่า

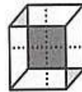
โดยทั้ง 4 คำสั่งนี้จะใช้ประสมกับตัวแปร char และ int เพื่อแจ้งให้ Compiler รับรู้ว่าตัวแปรที่ได้ประกาศไปนั้นมีคุณสมบัติอย่างไร

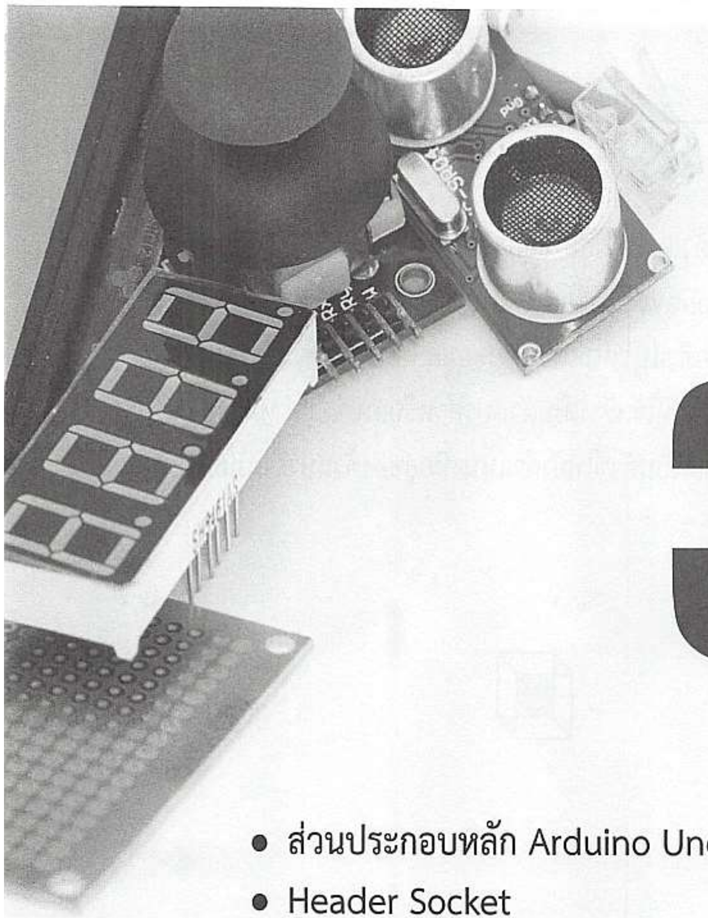
ชนิดตัวแปร	จำนวนบิต	ค่าข้อมูลที่เก็บได้
char	8	-128 ถึง +127
sign char	8	-128 ถึง +127
unsigned char	8	0 ถึง +255
int	16	-32768 ถึง +32767
signed int	16	-32768 ถึง +32767
unsigned int	16	0 ถึง 65535
short int	16	32768 ถึง +32767
signed short int	16	32768 ถึง +32767
unsigned short int	16	0 ถึง 65535
long int	32	-2147483648 ถึง +2147483647
Signed long int	32	-2147483648 ถึง +2147483647
unsigned long int	32	+ ถึง +4294967295
float	32	3.4E –38 ถึง 3.4E +38
double	64	1.7E –308 ถึง 1.7E +308
Long double	80	3.4E –4932 ถึง 3.4E +4932
void	0	ไม่มีค่า

 ชนิดของตัวแปรที่นิยมใช้กัน

- boolean ใช้เก็บค่าข้อมูล 2 แบบ คือ True (จริง) และ False (เท็จ)
- char ใช้เก็บค่ารหัสของตัวอักษร ซึ่งสามารถกำหนดเป็นค่าตัวเลข หรือตัวอักษรภายในขอบเขตของเครื่องหมายฟันเดียว '...' เช่น 'A' หรือ 0X41 หรือ 65
- byte ใช้เก็บค่าจำนวนเต็มแบบไม่คิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 256 ค่า (0-256)
- Int (Integer) ใช้เก็บค่าจำนวนเต็มแบบคิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 65536 ค่า คือ -32768 ถึง +32767
- unsigned int ใช้เก็บค่าจำนวนเต็มแบบคิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 65536 ค่า คือ 0-65536
- long ใช้เก็บค่าตัวเลขจำนวนเต็มแบบคิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 4294967296 คือ -2147483648 ถึง 2147483647
- unsigned long ใช้เก็บค่าตัวเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 4294967296 คือ 0-4294967295
- float ใช้เก็บค่าตัวเลขที่เป็นทศนิยมแบบคิดเครื่องหมาย ซึ่งสามารถเก็บค่าข้อมูลได้ 3.4E-38 ถึง 3.4E +38
- double ใช้เก็บค่าตัวเลขที่เป็นทศนิยมแบบคิดเครื่องหมายเช่นเดียวกับ float แต่มีค่าความละเอียดกว่าถึง 2 เท่า ซึ่งสามารถเก็บค่าได้มากถึง 1.7E -308 ถึง 1.7E +308
- void คือ ตัวแปรแบบที่ไม่มีการเก็บค่าใดๆ หมายถึง ไม่มีค่า
- arrays ใช้เก็บข้อมูลหลายค่าไว้ในตัวแปรเพียงชื่อเดียว ซึ่งจะมีตัวเลขสำหรับชี้ตำแหน่งการเก็บข้อมูลที่ต่างกัน โดยค่าลำดับของข้อมูลในตัวแปร arrays ตำแหน่งแรกจะมีค่าเป็น 0 เสมอ

- string คือ ตัวแปรที่ใช้เก็บข้อความ หรือตัวอักษรหลายตัว หมายถึง string คือ arrays ของตัวแปรแบบ char
- pointer คือตัวแปรที่ไม่ได้ใช้เก็บข้อมูล แต่ใช้เก็บค่าตำแหน่งที่อยู่ของหน่วยความจำที่ใช้สร้างเป็นตัวแปรสำหรับเก็บข้อมูล หมายถึง ตัวแปรนี้จะทำหน้าที่เป็นตัวชี้ไปยังตำแหน่งที่อยู่ของตัวแปรอื่นนั่นเอง





3

- ส่วนประกอบหลัก Arduino Uno R3
- Header Socket
- USB Socket
- LED On Board
- Reset Button

รู้จักกับบอร์ด Arduino Uno R3

ปัจจุบันบอร์ดไมโครคอนโทรลเลอร์ Arduino ได้พัฒนาไปถึงรุ่น Arduino Uno ซึ่งตอนนี้รุ่นใหม่ล่าสุด คือ Arduino Uno R3 หมายถึง บอร์ด Arduino รุ่นที่มีการแก้ไขปรับปรุงเป็นครั้งที่ 3 "Third Revision" โดยบอร์ด Arduino รุ่นดั้งเดิมที่คนไทยรู้จักกันดี คือ Arduino Uno R2 แต่วิวัฒนาการของ Arduino นั้นเริ่มต้นตั้งแต่ปี ค.ศ. 2005

สำหรับบอร์ด Arduino Uno R3 จะแบ่งย่อยออกได้เป็น 2 เวอร์ชัน ตามชนิดของชิปไมโครคอนโทรลเลอร์ คือ Arduino Uno R3 แบบใช้ชิป MCU แบบ DIP หรือ Dual Inline Package ซึ่งชิปไมโครคอนโทรลเลอร์ชนิดนี้จะสามารถถอดเปลี่ยนได้ โดยชิปจะเสียบเข้าไปใน Socket ที่บัดกรีติดกับบอร์ด ส่วนบอร์ด Arduino Uno R3 อีกเวอร์ชันหนึ่ง คือ Arduino Uno R3 SMD จะใช้ชิป MCU แบบ SMD หรือ Surface Mount Device สำหรับชิปไมโครคอนโทรลเลอร์ชนิดนี้ จะบัดกรีติดกับบอร์ด ซึ่งไม่สามารถถอดเปลี่ยนได้ และแน่นอนบอร์ด Arduino ทั้ง 2 เวอร์ชัน มีต้นทุนการผลิตที่ไม่เท่ากัน โดยชิปชนิดแบบไม่สามารถถอดเปลี่ยนได้ SMD จะมีต้นทุนที่ถูกกว่า ชิปชนิด DIP ที่สามารถถอดเปลี่ยนได้ แต่นอกเหนือจากนี้บอร์ด Arduino ทั้ง 2 เวอร์ชัน มีสเปกเหมือนกันทั้งหมด "ก็แค่บอร์ดเวอร์ชันที่แพงกว่า ถ้าเสียมาก็แค่ซื้อชิปมาเปลี่ยนใหม่ก็เท่านั้น"

HISTORY OF ARDUINO

2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017

Arduino was born out of the need for a low-cost microcontroller platform for Massimo Banzi's students at the **Interaction Design Institute Ivrea.**

It's named after a local pub:
Bar di Re Arduino.

The Arduino IDE (Integrated Development Environment) is built upon **Wiring** - a software project written by one of Banzi's students (**Hernando Barragán**). It provides easy-to-use libraries which hide some of the raw C++ going on behind the scenes.

Adafruit estimate 300,000 official boards produced

Tian
Industrial 101
Yun
Zero
101

Esplora
Robot

create.arduino.cc web based IDE is launched

Atmega328 again doubles the memory

Fio
Ethernet

First ever Arduino day 29/03/14

Uno Wi-Fi
MKRZero
Primo
MKR1000
MKRFOX

IDE revision 0001 released

Atmega168 doubles the flash memory

Atmega8 used for the first boards

LilyPad

LilyPad Snap
LilyPad USB
LilyPad Simple

Primo Core

Nano
Pro Mini
Pro

First 32-bit Arduino

IDE 1.8 released

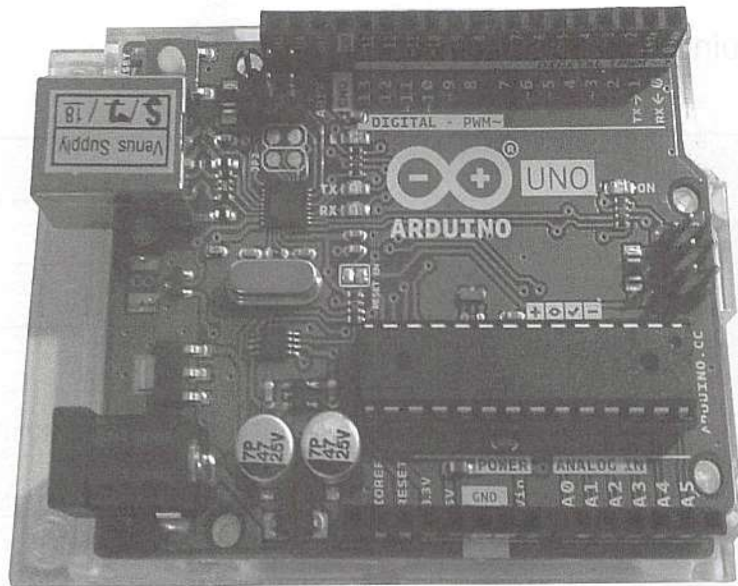
Arduino Begins

USB
NG
Uno
Leonardo
Due

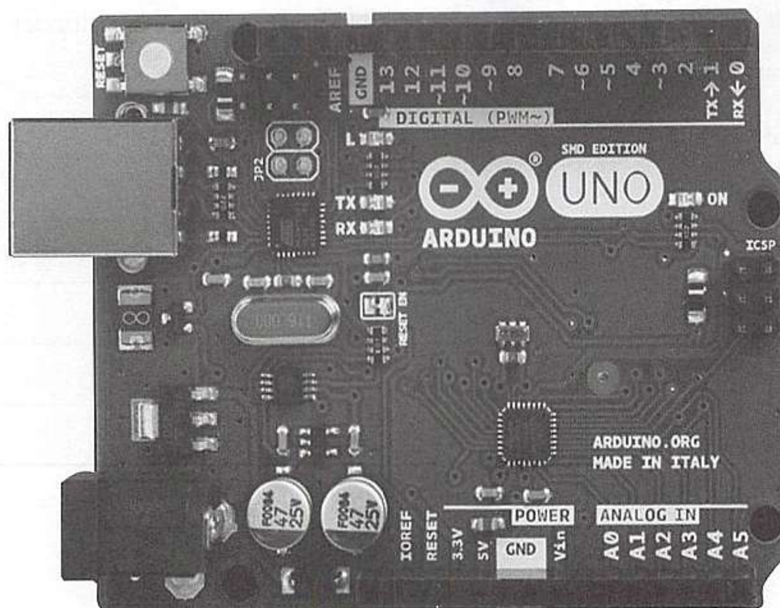
Arduino splits: arduino.cc (Genuino outside USA) and arduino.org

Arduino reunites under Arduino Foundation

Serial
Extreme
Decimila
Duemilanove
Mega
Mega 2560
Mega ADK
Micro



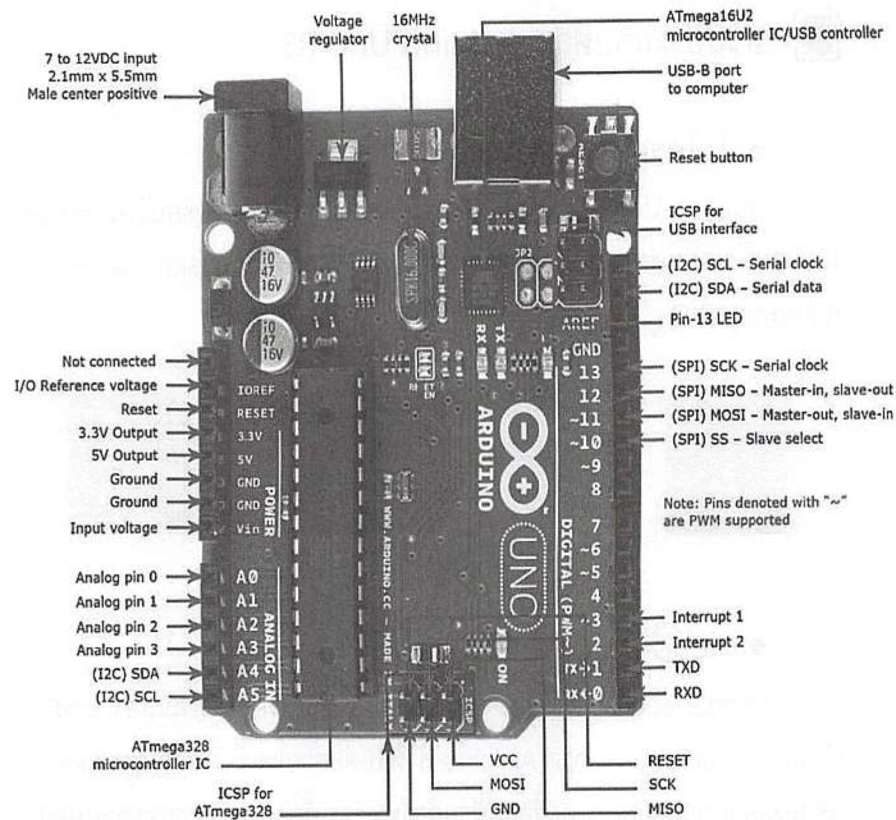
Arduino Uno R3 MCU DIP



Arduino Uno R3 MCU SMD

Arduino Uno R3 Specification

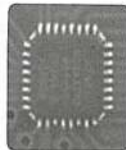
ไมโครคอนโทรลเลอร์	ATmega328P
แรงดันไฟฟ้าที่ใช้งาน	5V
แรงดันไฟฟ้าขาเข้า	ที่เหมาะสมที่สุด คือ 7-12V
ข้อจำกัดของแรงดันไฟฟ้าขาเข้า	6-20V
หมุด Digital I / O	14 หมุด (เป็นขาออกให้ PWM 6 หมุด)
หมุด PWM Digital I / O	6 หมุด
หมุด Input อะนาล็อก	6 หมุด
กระแสไฟ DC ต่อหมุด I / O	20 mA
หมุดกระแสไฟ DC สำหรับ 3.3V	50 mA
หน่วยความจำ Flash (Flash Memory)	32 KB (ATmega328P) ใช้ bootloader ขนาด 0.5 KB
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
ความยาว	68.6 มิลลิเมตร
ความกว้าง	53.4 มิลลิเมตร
น้ำหนัก	25 กรัม

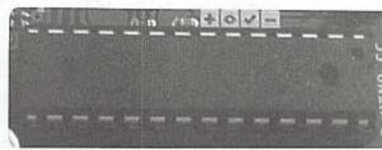

Voltage regulator
16MHz crystal
ATmega16U2 microcontroller IC/USB controller
7 to 12VDC input 2.1mm x 5.5mm Male center positive
USB-B port to computer
Reset button
ICSP for USB interface
(I2C) SCL - Serial clock
(I2C) SDA - Serial data
Pin-13 LED
Not connected
I/O Reference voltage
Reset
3.3V Output
5V Output
Ground
Ground
Input voltage
(SPI) SCK - Serial clock
(SPI) MISO - Master-in, slave-out
(SPI) MOSI - Master-out, slave-in
(SPI) SS - Slave select
Note: Pins denoted with "~" are PWM supported
Analog pin 0
Analog pin 1
Analog pin 2
Analog pin 3
(I2C) SDA
(I2C) SCL
Interrupt 1
Interrupt 2
TXD
RXD
ATmega328 microcontroller IC
ICSP for ATmega328
VCC
MOSI
GND
RESET
SCK
MISO
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
ON
ICSP

 ส่วนประกอบหลัก Arduino Uno R3

• ไมโครคอนโทรลเลอร์

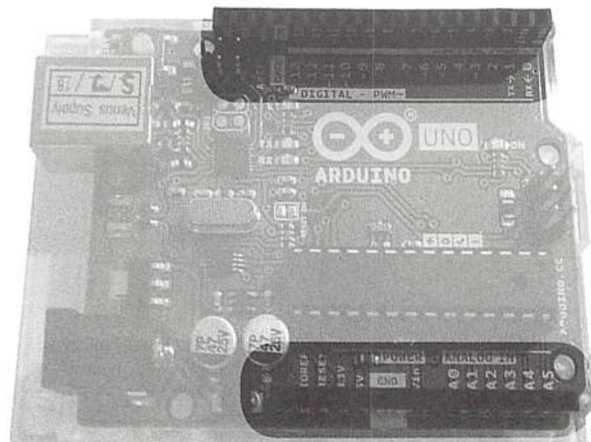
Arduino Uno R3 จะใช้ชิป Microcontroller ซึ่งผลิตโดยบริษัท Atmel โดยส่วนนี้จะเป็นเหมือนสมองของบอร์ดที่ใช้สำหรับการประมวลผล รวมถึงการควบคุมหมุด I/O





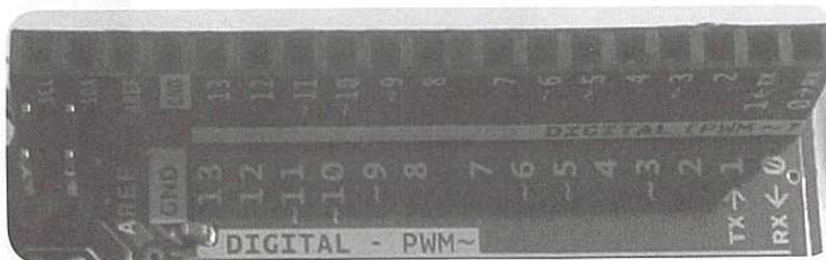
• Header Socket

Header Socket จะเชื่อมต่อมาจากขาของไมโครคอนโทรลเลอร์ ซึ่งจะเรียงอยู่บริเวณขอบของบอร์ด Arduino สำหรับต่อสายไฟเพื่อรับสัญญาณ Input หรือส่งสัญญาณ Output ทั้งนี้จะมีป้ายกำกับพร้อมทั้งหมายเลขหมุดปรากฏอยู่เพื่อความสะดวกต่อการใช้งาน โดย Header Socket จะแบ่งออกเป็น 3 กลุ่มหลัก คือ Digital Pin, Analog in Pin และ Power Pin



• Digital Pin

Digital Pin คือ หมุดสำหรับรับและส่งสัญญาณที่เป็น Digital โดยมี 2 สถานะ คือ On (0V) และ Off (5V)



• Analog in Pin



Analog in Pin คือ หมุดสำหรับรับค่าสัญญาณที่เป็น Analog

• Power Pin



Power Pin คือ หมุดสำหรับจ่ายแรงดันไฟฟ้าให้กับอุปกรณ์อิเล็กทรอนิกส์ หรือชุดโมดูลเซ็นเซอร์ต่างๆ ซึ่งจะมีทั้ง 5V และ 3.3V นอกจากนี้ยังมีหมุด Vin ที่จะให้ค่าความต่างศักย์เท่ากับไฟที่ต่อมาจากพอร์ตอะแดปเตอร์ (External Power Jack)



• USB Socket

USB Socket คือ พอร์ตที่ใช้ในการเชื่อมต่อสาย USB ในการ Upload Sketch Code ไปยังชิปไมโครคอนโทรลเอร์ รวมถึงจ่ายแรงดันไฟให้กับบอร์ด Arduino

• External Power Jack

External Power Jack คือ ช่องสำหรับเสียบ Power Adapter เพื่อต่อไฟจากแหล่งจ่ายภายนอก

• LED On Board

L LED



L LED คือ ไฟ LED ที่ติดอยู่กับบอร์ด ซึ่งเชื่อมต่อกับหมุด D13 ซึ่งจะใช้ในการทดสอบการทำงานของบอร์ด หรือใช้สำหรับ Sketch Code ที่กำหนดให้โมดูลทำงานแล้วไฟ LED ติด

Power LED

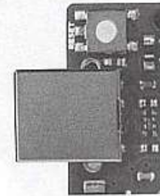


Power LED คือ ไฟ LED ที่ติดอยู่กับบอร์ด ซึ่งจะเป็นไฟแสดงสถานะ เมื่อไฟเข้าเลี้ยงบอร์ด เช่น หากไฟ LED ดวงนี้ขึ้นเป็นสีเขียว แสดงว่า บอร์ด Arduino พร้อมทำงานแล้ว

RX, TX LED

RX, TX LED คือ ไฟ LED ที่แสดงสถานะของการรับและส่งข้อมูล ในขณะที่บอร์ดกำลังทำการส่งหรือรับข้อมูล

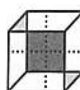




- **Reset Button**

Reset Button คือ ปุ่มสำหรับ Reset โปรแกรม Sketch Code ที่บอร์ด Arduino กำลังทำงาน เพื่อให้บอร์ดหยุดการทำงานเดิม จากนั้นจะเริ่มต้นทำงานใหม่ตั้งแต่บรรทัดแรก

หลายคนคงสงสัยว่า "ทำไมถึงมี Uno ยี่ห้ออื่นมากมาย" บอร์ดไมโครคอนโทรลเลอร์บางยี่ห้อ ตั้งชื่อรุ่นเป็น Uno R4 เลยก็มี อีกทั้งบอร์ดยี่ห้ออื่นๆ "ที่ไม่ใช่ Arduino" ยังมีรูปร่างและลักษณะเหมือนกันกับบอร์ด Arduino ด้วยเป็นเพราะบอร์ดไมโครคอนโทรลเลอร์ เป็นแฟลตฟอร์มแบบ Open-Source เช่นนั้นลายวงจร รวมถึง PCB Design ของบอร์ด Arduino "ไม่มีลิขสิทธิ์" และถูกเปิดเผยหมดจนหมดเปลือก เพื่อเปิดโอกาสให้คนทั้งโลกสามารถดาวน์โหลดแบบและลายวงจร "Copy" ไปทำซ้ำผลิตออกจำหน่ายได้โดยไม่ผิดลิขสิทธ์ จึงทำให้ผู้ผลิตมากมายทั่วโลกมีการผลิตบอร์ดเหมือนหรือคล้ายกับบอร์ด Arduino ออกมาจำหน่ายมากมาย อาจกล่าวได้ว่า เป็นบอร์ดไมโครคอนโทรลเลอร์ เลียนแบบบอร์ด Arduino ก็ว่าได้ แต่ก็กล่าวไม่ได้เต็มปาก เพราะบอร์ดไมโครคอนโทรลเลอร์ Arduino แท้ๆ ที่ผลิตจาก Italy ก็ก๊อบปี้แบบและลายวงจร PCB Design มาเหมือนกัน "เพราะแฟลตฟอร์มเป็น Open-Source ไง" โดยจะเรียกบอร์ดฝาแฝดเหล่านี้ว่าบอร์ด Compatible แต่ความแตกต่างนั้น เพียงแค่ไม่ได้ผลิตมาจากโรงงานของ Arduino โดยตรงเท่านั้นเอง แต่ Specification และประสิทธิภาพในการทำงานอาจเหมือนกันทุกประการ "ก็แล้วแต่ยี่ห้อนะ" และที่สำคัญบอร์ดไมโครคอนโทรลเลอร์ Compatible มีราคาที่ถูกกว่ามากอีกด้วย





4

- การดาวน์โหลดโปรแกรม Arduino IDE
- การเชื่อมต่อบอร์ด Arduino กับระบบปฏิบัติการคอมพิวเตอร์
- ปัญหาตายน้ำตื้นของมือใหม่
- การ Verify (Compile) และการ Upload โปรแกรมไปยังบอร์ด Arduino

ติดตั้งและลองใช้ Arduino IDE

ไม่ว่าประสบการณ์เรากับ Arduino จะเป็นอย่างไร มากน้อยแค่ไหน "ปวดกระบอกตา ไม่เข้าใจอะไรเลย" หากได้เปิดมาถึงบทนี้ จงปล่อยวางทุกสิ่งทุกอย่างที่ไม่เข้าใจ "อะไรของมันวะ ทำไมยังไงก็ตัน" แล้วทำตามขั้นตอนในบทนี้เพื่อเริ่มต้นการใช้งาน Arduino แล้วเราจะสนุกกับบอร์ด Arduino และสามารถเข้าใจอะไรหลายสิ่งหลายอย่างพร้อมกันไปด้วยเลย

บางคนฝันเอาไว้ว่า "ฉันจะทำเซ็นเซอร์กันขโมยไว้ประตูหน้าบ้าน โดยใช้บอร์ด Arduino" พอเริ่มต้นทำตามที่ฝัน "ซื้อชุดอุปกรณ์ Arduino มาพร้อมแล้ว" แต่แค่เริ่มต้นพลังชีวิตก็หมดไปกับการเขียนภาษา C การตรวจสอบเงื่อนไข คำสั่ง if คำสั่ง while คำสั่ง continue บลา บลา บลา... จับบอร์ด Arduino หงายดูด้านบน แล้วก็หงายดูด้านล่าง จับอุปกรณ์นั้นขึ้นมาดู จับชุดโมดูลเซ็นเซอร์ขึ้นมาอ่านตัวเลขเล็กๆ ในตัวถัง ทำได้เท่านี้มากว่า 3 วัน ตามองเหลือบไปเห็นตำรา Arduino ที่วางอยู่ข้างๆ "ก็ถึงกับถอนหายใจเฮือกใหญ่" ไม่มีอะไรคืบหน้า "นี่ Goo ซื้ออะไรมาวะเนี๊ย" หมดกำลังใจที่จะสานต่อ ความพยายามเหือดหาย เช่นนั้นทำไมเราไม่โยนตำรา 500 กว่าหน้าทิ้ง แล้วเปิดหนังสือเล่มนี้ หรือไม่ก็พิมพ์คีย์เวิร์ดคำว่า "Arduino Startup" บนช่องค้นหาของ Google หรือ YouTube เห็นอะไรอยู่ตรงหน้า ก็ลองทำตามเขา ลองต่อตามเขา ลองคลิกตามเขา ลองพิมพ์

เหมือนเขา ลองทำตามเขาทั้งหมด ถ้าสิ่งที่ต้องลองทำตามเขาครั้งนี้สำเร็จ ครั้งต่อไปก็ย่อมจะสำเร็จเช่นเคย ไม่แน่ครั้งต่อไปอาจดัดแปลงชุดคำสั่งของคนอื่น เพื่อให้โปรแกรมทำงานตามที่เราต้องการก็ได้ "เท่านี้เอง ทฤษฎีภาษา C ผีปอบ ค่อยไปว่ากันทีหลังเถอะ"

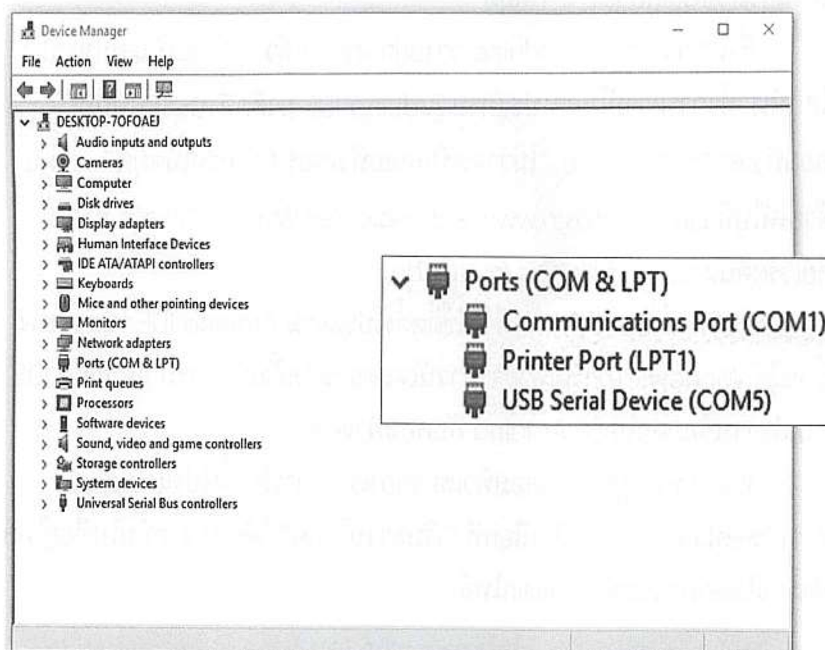
เพราะบางครั้งการมัวหมกมุ่นอ่านแต่ทฤษฎี ไม่ได้ช่วยให้อะไรดีขึ้น "ยิ่งอ่านยิ่งไม่เข้าใจ ยิ่งอ่านทวนซ้ำในประโยคที่สงสัยและไม่รู้เรื่อง ยิ่งมึนตึ้บปวดหัวไปกันใหญ่" ถ้าเช่นนั้นมาลองปฏิบัติจริงกันเลยดีกว่า "ตัดสินใจไปซื้อชุด Arduino Combo Set" มาเลย ไม่ต้องห่วงไม่ต้องกลัวว่า จะใช้ไม่ได้ใช้ไม่เป็น บางครั้งการทำอะไรตามตัวอย่างที่สำเร็จ "พิสูจน์แล้ว" ก็ทำให้เรามีกำลังใจที่จะพยายามทำความเข้าใจกับอะไรก็ตามที่เราคิดว่า "Goo ไม่ได้เกิดมาเพื่อสิ่งนี้" แล้วความสำเร็จจะรออยู่ตรงหน้าเรา เพียงแต่ตอนนี้เรามองไม่เห็นความสำเร็จเท่านั้นเอง "จนกว่าเราจะลงมือทำ"

 Arduino IDE

Arduino IDE คือ โปรแกรมที่ใช้สร้าง Sketch Code และ Upload Sketch Code ไปยังบอร์ด Arduino โดยโปรแกรม Arduino IDE จะต้องทำการติดตั้งบนระบบปฏิบัติการคอมพิวเตอร์ และทำการเชื่อมต่อบอร์ด Arduino กับ USB Port ของเครื่องคอมพิวเตอร์ โดยเมื่อทำการเขียน Sketch Code บนโปแกรม Arduino IDE เสร็จเรียบร้อย จะต้องทำการ Upload โปรแกรมชุดคำสั่ง (Sketch Code) ไปยังบอร์ด Arduino เพื่อให้มีการส่งสัญญาณไปยังชุดโมดูลอิเล็กทรอนิกส์

ข้อสำคัญในการเชื่อมต่อบอร์ด Arduino กับโปรแกรม Arduino IDE บนระบบปฏิบัติการคอมพิวเตอร์

1. การค้นหา Port เพื่อให้ระบบปฏิบัติการคอมพิวเตอร์ค้นหาเจอ และเชื่อมต่อกับบอร์ด Arduino
2. การเลือก Port ในโปรแกรม Arduino IDE จะต้องตรงกันกับ Port ของเครื่องคอมพิวเตอร์ที่เชื่อมต่อกับบอร์ด Arduino "พอร์ต USB ที่เสียบกับเครื่อง ระบบปฏิบัติการเรียกพอร์ตนั้นว่าอะไร ก็เลือกตามที่ระบบตั้งชื่อ"
3. หากไม่สามารถเชื่อมต่อกับบอร์ด Arduino ไม่ว่าทางใด ให้ไปที่ Control Panel และไปที่ Device Manager จากนั้นทำตามขั้นตอนในหัวข้อย่อย "ปัญหาตายน้ำตื่นของมือใหม่"



 การดาวน์โหลด Arduino IDE

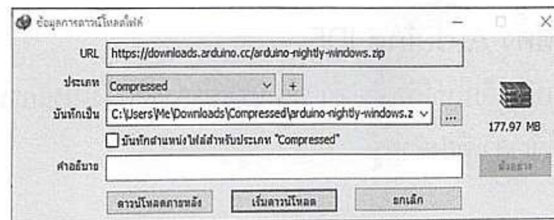
โปรแกรม Arduino IDE คือ เครื่องมือการเขียนโปรแกรมที่ใช้งานกับ Arduino ทุกรุ่น และรวมถึงบอร์ดไมโครคอนโทรลเลอร์ยี่ห้ออื่นๆ ที่ใช้ชิปไมโครคอนโทรลเลอร์ตระกูล AVR โดยโปรแกรมจะมีเครื่องมือสำหรับการเชื่อมต่อกับบอร์ด Arduino เช่น การค้นหา Arduino ที่มีการเชื่อมต่อกับเครื่องคอมพิวเตอร์ หรือการเลือกรุ่น Arduino ที่เชื่อมต่ออยู่เพื่อตรวจสอบขนาดของโปรแกรมที่จะเขียนหรือสร้างขึ้น นอกจากนี้ยังไลบรารีต่างๆ ที่รองรับกับ Arduino ในแต่ละรุ่น อีกทั้งยังมีโปรแกรมที่ทำการตรวจสอบและเชื่อมต่อผ่านซีเรียลนัมเบอร์โดยตรงสำหรับระบบปฏิบัติการคอมพิวเตอร์ด้วย

โดยตัวอย่างการดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE ของหนังสือเล่มนี้ จะใช้ระบบปฏิบัติการ Windows 10 ซึ่งถือว่ามีความซับพอร์ตโปรแกรม Arduino IDE มากที่สุด

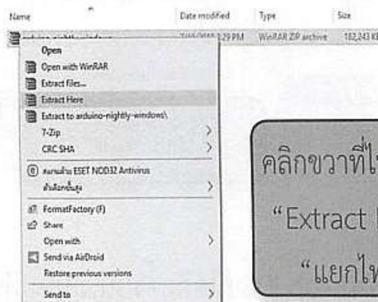
ซึ่งหากเรามีบอร์ด Arduino วางอยู่ตรงหน้าแล้ว ไม่ว่าจะเป็นรุ่นไหนปีใด ให้เราทำการดาวน์โหลดโปรแกรม Arduino IDE มาติดตั้งบนระบบปฏิบัติการคอมพิวเตอร์ของเราได้เลย “ไม่ว่าจะเป็นคอมพิวเตอร์ PC หรือโน้ตบุ๊ค” เริ่มต้นด้วยให้ไปที่ Link นี้ https://www.arduino.cc/en/Main/Software หรือใช้คีย์เวิร์ดค้นหาใน Google ว่า arduino ide

สำหรับระบบปฏิบัติการที่จะใช้ติดตั้งโปรแกรม Arduino IDE ในตัวอย่างนี้ จะใช้ Windows 10 เพื่อทำการดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE รวมถึงการเชื่อมต่อบอร์ด Arduino กับคอมพิวเตอร์

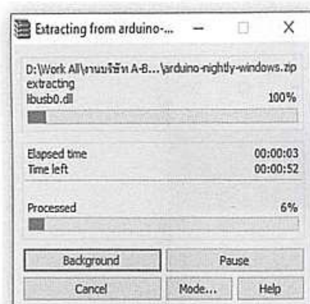
หากระบบปฏิบัติการคอมพิวเตอร์ของเรา มีการติดตั้งโปรแกรมช่วยดาวน์โหลดไว้ ก็สามารถคลิกเลือกที่ “เริ่มดาวน์โหลด” ได้เลย จากนั้นไปที่อยู่ไฟล์ที่ดาวน์โหลดมา เพื่อทำการแตกไฟล์



Name	Date modified	Type	Size
arduino-nightly-windows	7/10/2018 3:29 PM	WinRAR ZIP archive	182,243 KB



คลิกขวาที่ไฟล์แล้วเลือก “Extract Here” หรือ “แยกไฟล์ไว้ที่นี่”



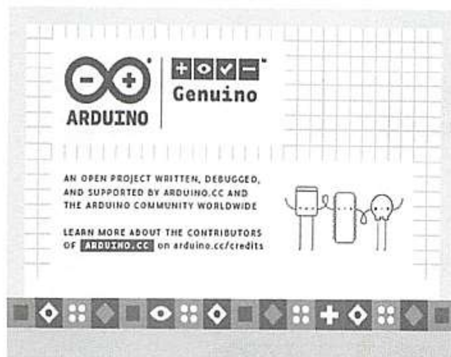
โฟลเดอร์ที่แยกไฟล์ออกมาเรียบร้อยแล้ว ดับเบิลคลิกเข้าไปได้เลย

Name	Date modified	Type	Size
arduino-nightly	6/18/2018 3:33 PM	File folder	
arduino-nightly-windows	7/10/2018 3:29 PM	WinRAR ZIP archive	182,243 KB

ให้ทำการดับเบิลคลิกที่ไฟล์ arduino ที่มีนามสกุล .exe หรือ Application เพื่อทำการติดตั้งโปรแกรม Arduino IDE ลงบนปฏิบัติการคอมพิวเตอร์

 การติดตั้ง Arduino IDE

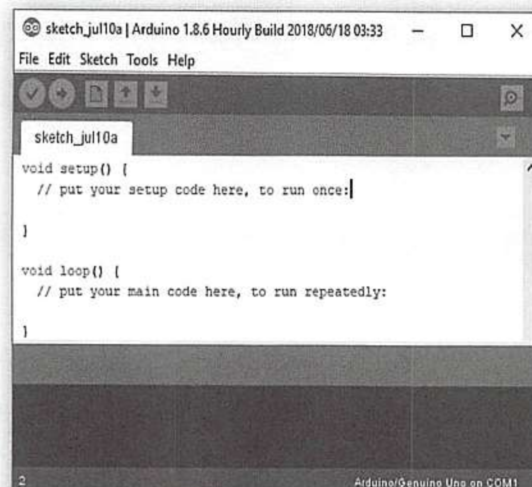
เมื่อดับเบิลคลิกไฟล์ติดตั้งแล้ว ระบบจะทำการติดตั้งโปรแกรม Arduino IDE ทันที โดยใช้เวลาเพียงชั่วครู่



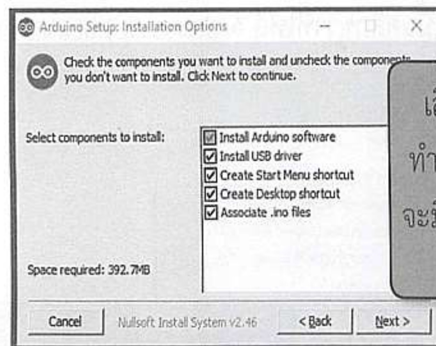


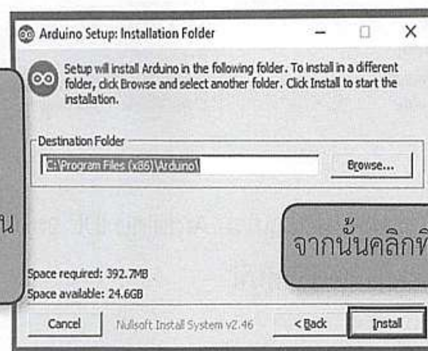
คลิกที่ Allow access

เพียงเท่านี้โปรแกรม Arduino IDE ก็ได้ติดตั้งบนระบบปฏิบัติการคอมพิวเตอร์แล้ว “ง่ายมั้ย”



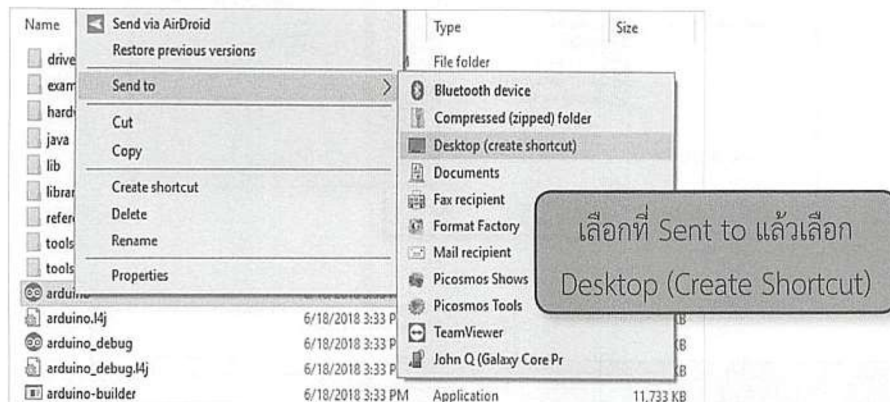
อันที่จริงถ้าเป็นโปรแกรม Arduino IDE เวอร์ชันเก่า ระบบจะต้องติดตั้งแบบ Arduino Setup : Installation Option ซึ่งจะต้องมีขั้นตอนในการ Setup โปรแกรมซึ่งก็ไม่ยุ่งยากซับซ้อนอะไร





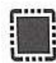


สำหรับไอคอนเพื่อใช้ในการเปิดโปรแกรม หากเป็นการติดตั้งโปรแกรม Arduino IDE รุ่นใหม่ “Arduino Genuino Uno” จะไม่ปรากฏไอคอนที่หน้า Desktop เช่นนั้นเราจะต้องทำการ “Desktop (Create Shortcut)” ให้ไอคอนโปรแกรมแสดงที่หน้า Desktop เพื่อสะดวกในการเข้าใช้งานโปรแกรม โดยให้เข้าไปที่โฟลเดอร์ Arduino จากนั้นคลิกขวาที่ไฟล์ Arduino.Application (.exe)



ไอคอนโปรแกรม Arduino IDE จะไปปรากฏที่หน้า Desktop ของเราทันที



 การเชื่อมต่อบอร์ด Arduino กับระบบปฏิบัติการคอมพิวเตอร์

ในบทก่อนเราได้ทำความรู้จักกับบอร์ด Arduino Genuino Uno กันไปแล้ว เมื่อถึงขั้นตอนนี้ให้เราเตรียมความพร้อมของอุปกรณ์สำหรับการ Startup เพื่อเชื่อมต่อบอร์ด Arduino กับระบบปฏิบัติการคอมพิวเตอร์

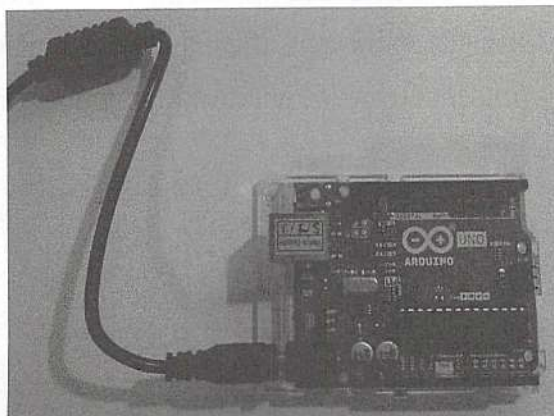
สาย USB Link Cable และ Power Supply 9V 2A



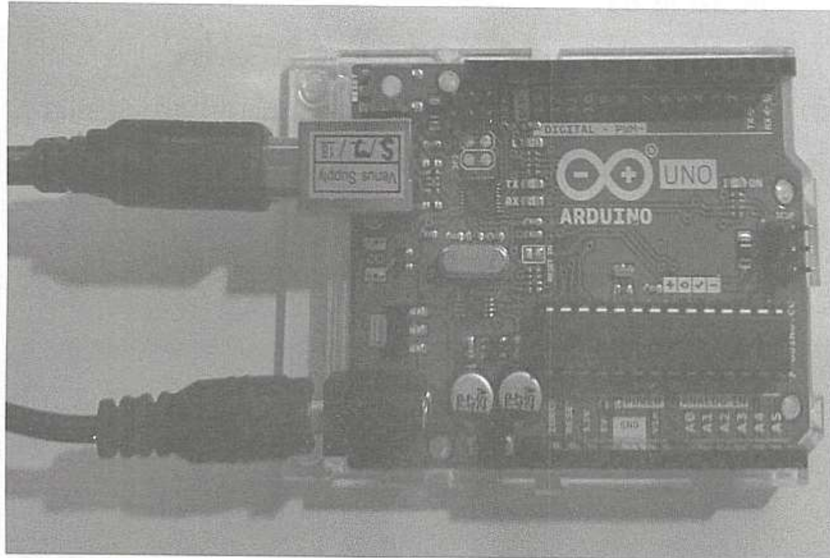


การต่อสาย USB และการจ่ายไฟเข้าบอร์ด Arduino

1. เริ่มต้นด้วยการต่อสายอะแดปเตอร์ (Power Supply) เข้าที่บอร์ด Arduino ดันเข้าตรงๆ เบาๆ อย่าโยก อย่าหมุน อย่าแรง

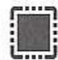


2. ตามด้วยต่อสาย USB เข้ากับบอร์ด Arduino โดยใช้หัวด้านหกเหลี่ยม

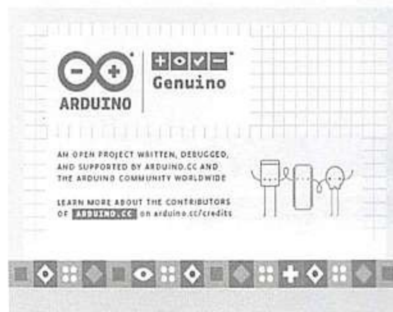


3. จากนั้นต่อสาย USB เข้ากับพอร์ตเครื่องคอมพิวเตอร์ โดยใช้หัวด้านสี่เหลี่ยม (เมื่อต่อสาย USB แล้ว หากเป็นบอร์ด Arduino รุ่นใหม่ ระบบจะ Auto Run ติดตั้ง Driver ของบอร์ด Arduino ทันที)

โดยปกติเมื่อเชื่อมต่อสาย USB จากบอร์ด Arduino เข้าเครื่องคอมพิวเตอร์แล้ว ไฟ LED ON บนบอร์ด Arduino จะเปล่งแสงสีเขียวขึ้นมาทันที เนื่องจากบอร์ดได้รับแรงดันไฟฟ้าจาก Power Supply ที่ประมาณแรงดัน 5V กระแส 500 mA จากเครื่องคอมพิวเตอร์ แต่แน่นอนไฟฟ้าเพียงเท่านี้ ไม่เพียงพอต่อการทำงานของไมโครคอนโทรลเลอร์ เช่นนั้นให้ทำการต่อหัวอะแดปเตอร์เข้ากับปลั๊กไฟบ้านได้เลย

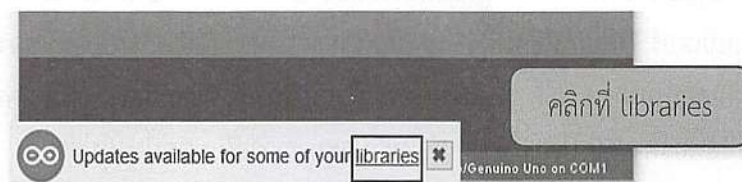
 Arduino Getting Started

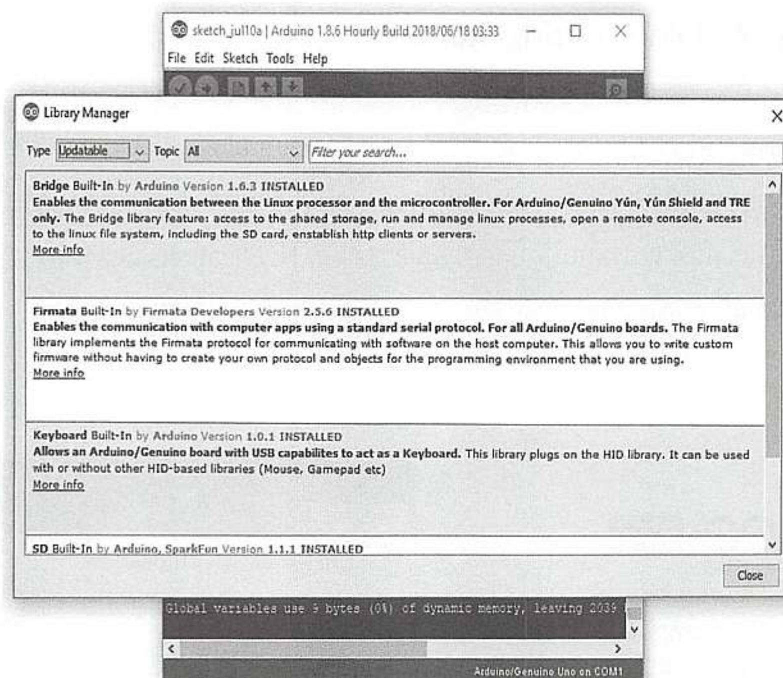
ถ้าเราใช้บอร์ด Arduino รุ่นใหม่ หรือ Arduino Genuino Uno หรือ Arduino Uno R3 ขึ้นไป เมื่อทำการต่อสาย USB จากบอร์ด Arduino เชื่อมต่อกับเครื่องคอมพิวเตอร์แล้ว ระบบจะทำการ Auto Run ติดตั้ง Driver ของบอร์ด Arduino ทันที โดยระบบปฏิบัติการจะมีการ New Notifications แสดงให้เราทราบทันที เช่นนั้นให้เราเปิดโปรแกรม Arduino IDE ขึ้นมาได้เลย





จากนั้นระบบจะให้เราทำการอัพเดต Libraries เพื่อให้คลังอุปกรณ์ต่างๆ เป็นปัจจุบันสำหรับการใช้โปรแกรม Arduino IDE โดยเฉพาะหากมีการเพิ่ม Libraries เข้าไปในโฟลเดอร์ Libraries ของโปรแกรม ระบบจะให้เราอัพเดตทันที



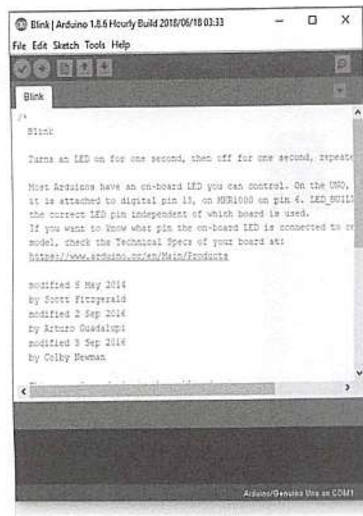


 ปัญหาตายน้ำตื้นของมือใหม่

ขั้นตอนนี้ทำเอาคนหัวเสียกันมาเยอะ เพราะไม่สามารถอัพโหลดโปรแกรมลงบอร์ด Arduino เพราะเลือก Serial Plotter ไม่ตรงกับพอร์ตที่เชื่อมต่อบอร์ด Arduino แล้วก็คิดไปต่างๆ นานา บอร์ดเสียหรือเปล่า "ก็พึ่งแกะกล่อง" คอมฯ เสียหรือเปล่า "ก็ใช้อยู่ทุกวัน" พอร์ต USB เสียหรือเปล่า "พอเสียบปุ๊บ notifications ก็ขึ้นปั๊บ" อ้าว...! แล้วมันเป็น...?

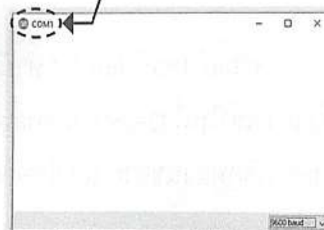
เครื่องคอมพิวเตอร์บางคนมีพอร์ตเสียบ USB มากกว่า 5 พอร์ต แถมมีที่จอมอนิเตอร์ "เสียบได้อีกหนึ่ง" เช่นนั้นเราจะต้องหาว่าบอร์ด Arduino ของเราเชื่อมต่อกับพอร์ต USB ของเครื่องคอมพิวเตอร์ที่พอร์ตไหน แล้วคอมพิวเตอร์ลำดับชื่อพอร์ต USB ที่เราเสียบที่เท่าไหร่ ประมาณว่าเสียบตรงไหนไม่สำคัญ สำคัญที่เราเลือกชื่อพอร์ต USB ในโปรแกรม Arduino IDE ว่าอะไร "" พอร์ตที่เท่าไหร่ของเครื่องคอมพิวเตอร์ "COM1, COM2, ... COM6"

โดยปกติถ้าเราใช้บอร์ด Arduino รุ่น Arduino Uno R3 ขึ้นไป ระบบโปรแกรม Arduino IDE จะทำการค้นหาให้เสร็จสรรพ ไม่ต้องทำอะไรเลย แค่เลือกลำดับพอร์ตให้ตรงกับที่เราเสียบสาย USB โดย Plotter จะมีชื่อว่า COM และตามด้วยลำดับตัวเลขตามที่ระบบปฏิบัติการคอมพิวเตอร์ได้กำหนดไว้ “COM1, COM2, COM3 ...” สำหรับระบบปฏิบัติการ Windows 10 กับบอร์ด Arduino Uno R3 รู้สึกว่าจะซับพอร์ตกันได้อย่างดี ทำให้การค้นหา Serial Plotter ได้อย่างรวดเร็ว แทบจะกล่าวได้ว่า “อัตโนมัติ”

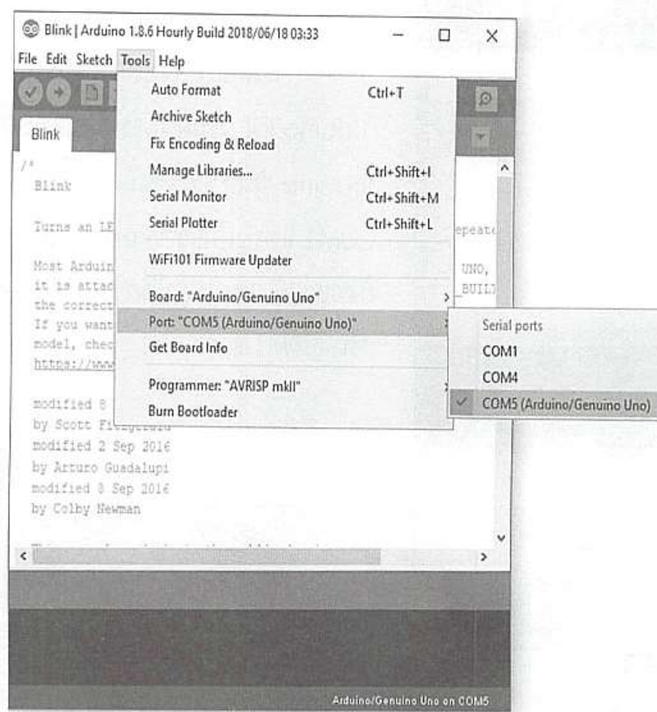


การเลือกพอร์ตในโปรแกรม Arduino IDE เริ่มต้นการเชื่อมต่อบอร์ด Arduino ระบบจะกำหนดค่าพื้นฐานไว้ที่ COM1 แต่บังเอิญว่าเราอาจต่อสาย USB ยังพอร์ตที่ระบบปฏิบัติการไม่ได้กำหนดให้เป็น COM1



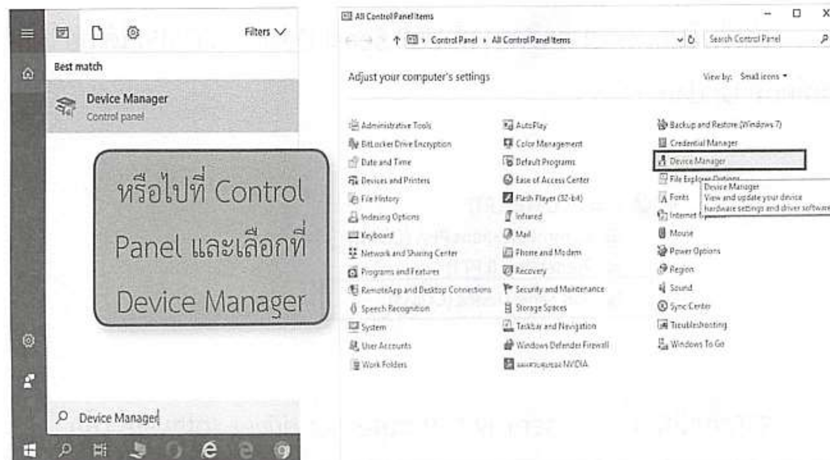


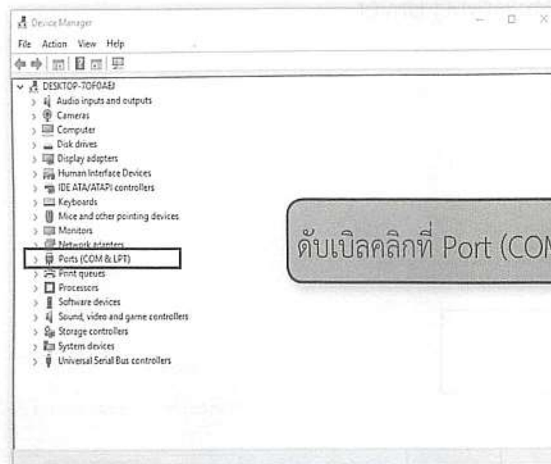
ดังนั้นเราจะต้องเลือกลำดับพอร์ตที่ถูกต้อง โดยเข้าไปที่ Tools แล้วเลือกที่ Port ซึ่งโดยปกติแล้ว หากเป็นบอร์ด Arduino Genuino Uno ระบบจะค้นหาเจอแล้ว และแสดงให้เรารู้ว่า "บอร์ด Arduino อยู่พอร์ตลำดับไหน" เราก็เพียงเลือกพอร์ตลำดับนั้น ซึ่งชื่อพอร์ตที่ถูกต้อง จะมีชื่อบอร์ดตามหลังชื่อพอร์ตแล้ว จากตัวอย่างระบบแสดงเป็น COM5 (Arduino/Genuino Uno) นั่นแสดงว่า เราต่อสาย USB เข้าพอร์ตเครื่องคอมพิวเตอร์ลำดับที่ 5 ถ้าอยากจะไล่ว่าพอร์ตไหนต้นพอร์ตไหนท้าย ให้ไล่มาจากพอร์ตที่ออนบอร์ด CPU (พอร์ต USB ที่ติดกับบอร์ด CPU)



หรือถ้าใครมีปัญหา "ทำยังไงโปรแกรมก็หาบอร์ดไม่เจอ" ให้แก้ปัญหาด้วยการเข้าไปที่ Device Manager "ไม่รู้อีกว่ามันอยู่ที่ไหนของ Windows" ง่ายๆ คลิกที่แว่นขยาย แล้วใช้คีย์เวิร์ดนี้ Device Manager แล้วจะเจอทางเข้าเอง





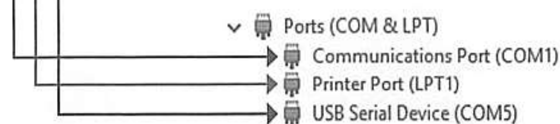


สังเกตดูที่ชื่อพอร์ตที่มีการเชื่อมต่อกับเครื่องคอมพิวเตอร์

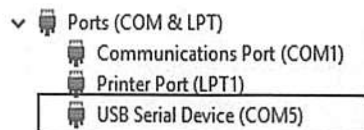
COM1 คือ การเชื่อมต่อกับการสื่อสาร

LPT1 คือ การเชื่อมต่อกับเครื่องพิมพ์

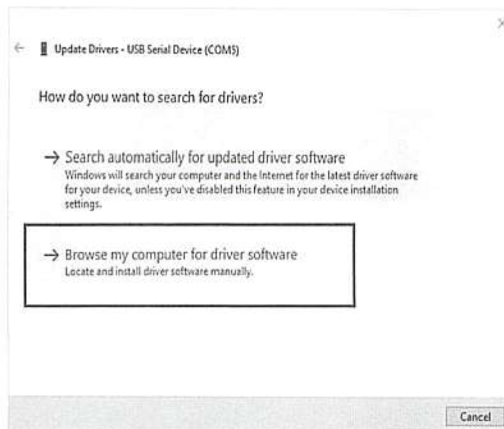
COM5 คือ เครื่องอุปกรณ์ที่เชื่อมต่อกัน (ซึ่งก็น่าจะเป็นพอร์ตนี้)



ถ้าเช่นนั้นก็ให้ทำการคลิกขวาที่ USB Serial Device (COM5) ได้เลย โดยให้เลือกที่ Update Driver

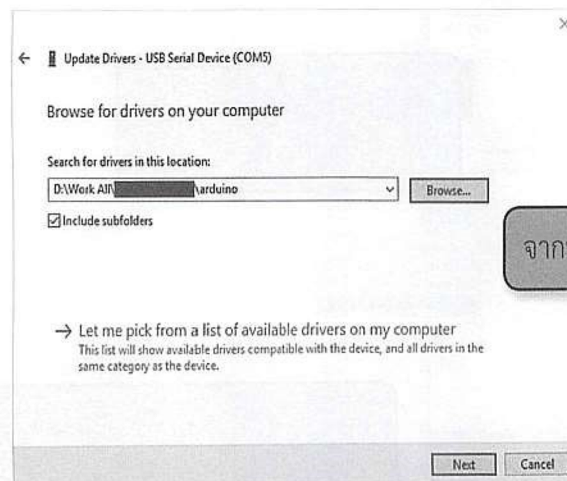


จากนั้นเลือกที่ Browser my computer for driver software เพื่อระบบปฏิบัติการทำการค้นหา Arduino และเชื่อมต่อกับระบบโปรแกรม Arduino IDE รวมถึงการติดตั้ง Driver





เลือกโฟลเดอร์ที่เก็บไฟล์ Arduino (ที่ได้ทำการดาวน์โหลดมาติดตั้งก่อนหน้านี้) แล้วคลิกที่ OK

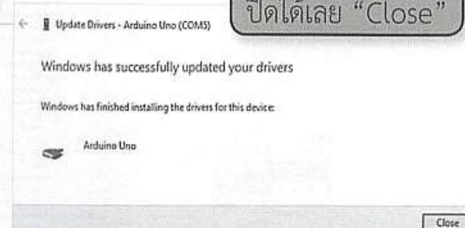


จากนั้นคลิกที่ Next

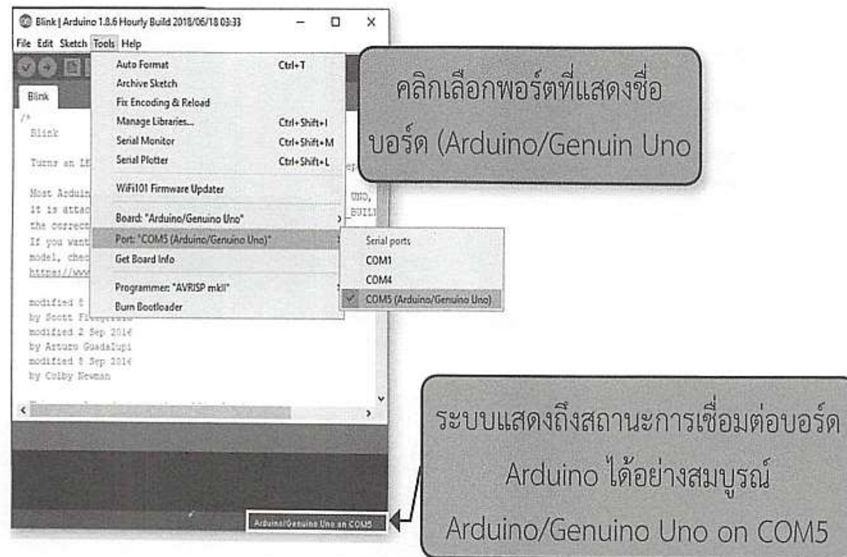
เลือกที่ Install เพื่อให้ระบบทำการติดตั้ง



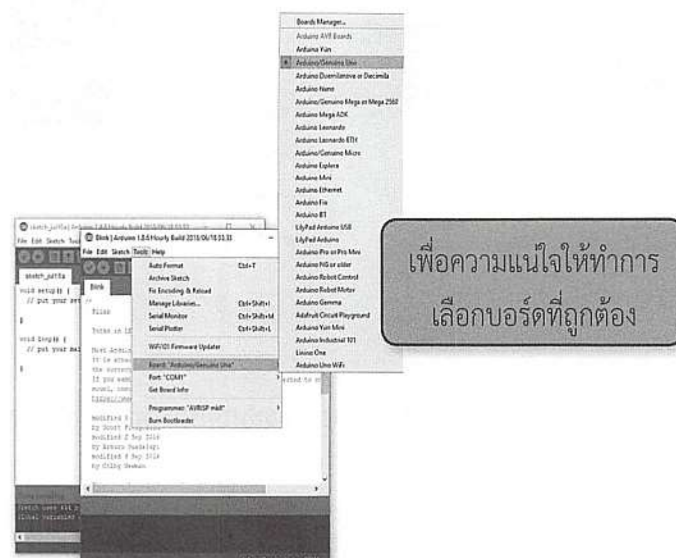
เมื่อระบบทำการอัพเดต Driver เรียบร้อยแล้ว ก็กดปิดได้เลย "Close"



จากนั้นลองไปตรวจสอบดูที่ Tools ว่า ระบบค้นหาพอร์ตที่เชื่อมต่อกับบอร์ด Arduino ได้แล้วหรือไม่



โดยปกติเมื่อทำการ Update Driver แล้ว เมื่อระบบเจอพอร์ตที่เชื่อมต่อกับบอร์ด Arduino แล้ว โปรแกรม Arduino IDE จะแสดงการเชื่อมต่อโดยอัตโนมัติ



สำหรับใครที่ทำยังไงระบบโปรแกรม Arduino IDE ก็ไม่สามารถเชื่อมต่อกับบอร์ด Arduino ได้สักที ให้ไปที่โฟลเดอร์ที่เก็บไฟล์ Arduino (ที่ได้ทำการดาวน์โหลดมาติดตั้งก่อนหน้านี้) จากนั้นให้ไปที่โฟลเดอร์ย่อย Driver

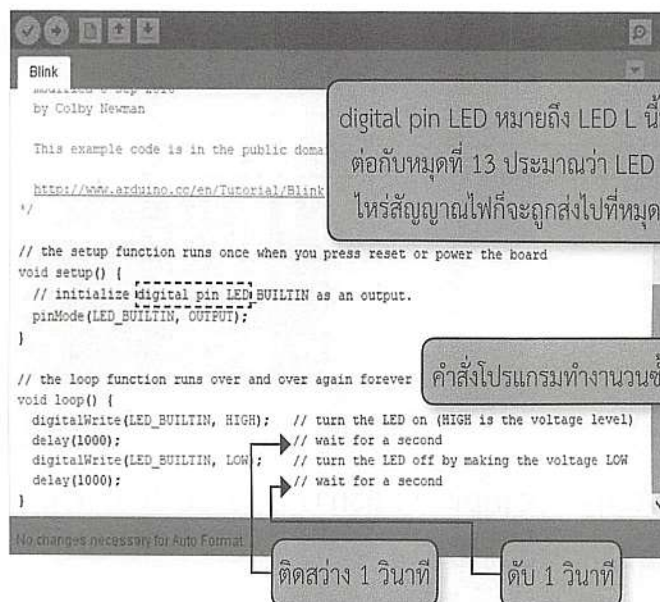
Name	Date modified	Type	Size
amd64	6/18/2018 3:33 PM	File folder	
CP210x_6.7	6/18/2018 3:33 PM	File folder	
CP210x_6.7.4	6/18/2018 3:33 PM	File folder	
FTDI USB Drivers	6/18/2018 3:33 PM	File folder	
ia64	6/18/2018 3:33 PM	File folder	
license	6/18/2018 3:33 PM	File folder	
x86	6/18/2018 3:33 PM	File folder	
AdafruitCircuitPlayground	6/18/2018 3:33 PM	Security Catalog	10 KB
AdafruitCircuitPlayground	6/18/2018 3:33 PM	Setup Information	4 KB
arduino	6/18/2018 3:33 PM	Security Catalog	11 KB
arduino	6/18/2018 3:33 PM	Setup Information	10 KB
arduino_gemma	6/18/2018 3:33 PM	Security Catalog	11 KB
arduino_gemma	6/18/2018 3:33 PM	Se	
arduino-org	6/18/2018 3:33 PM	Se	
arduino-org	6/18/2018 3:33 PM	Se	
dpinst-amd64	6/18/2018 3:33 PM	Ap	
dpinst-x86	6/18/2018 3:33 PM	Ap	
genuino	6/18/2018 3:33 PM	Se	
genuino	6/18/2018 3:33 PM	Se	
linino	6/18/2018 3:33 PM	Setup Information	4 KB

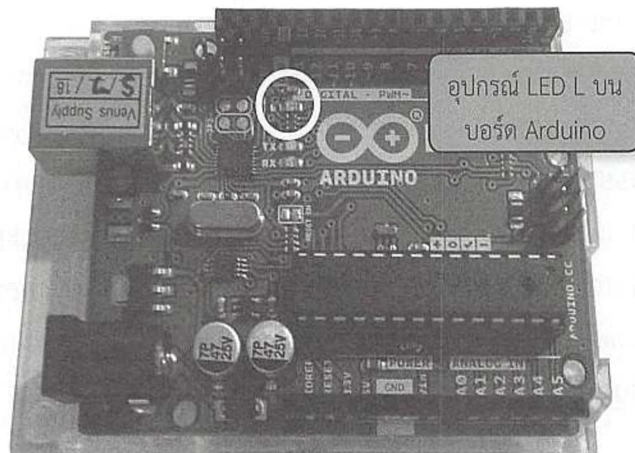
แล้วทำการดับเบิลคลิกที่
ไฟล์ arduino.Setup (.inf)
แล้วให้ระบบจัดการกันเอง

 การ Verify (Compile) และการ Upload โปรแกรมไปยังบอร์ด Arduino

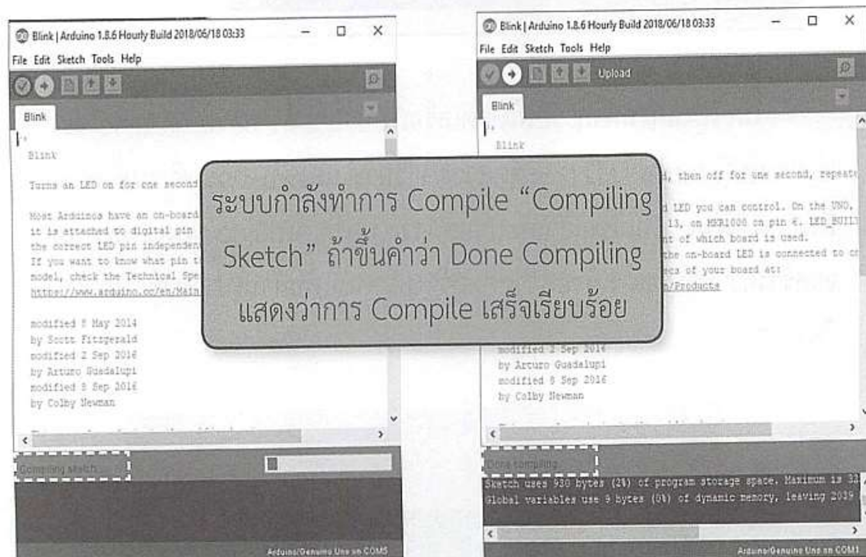
"Arduino First Project โปรแกรมแรกของฉัน" ในตอนเริ่มต้น เราได้ทำการเปิดไฟล์ Sketch Code ซึ่งเป็นโปรแกรมพื้นฐาน หรือ Blink อันเป็นโปรแกรมชุดคำสั่งเบื้องต้นที่จะใช้ตรวจสอบว่า บอร์ด Arduino มีสภาพการใช้งานเป็นปกติหรือไม่ หรือโดยปกติแล้ว นักเขียนโปรแกรม Arduino หากเกิดปัญหาบางอย่าง เช่น โมดูลเซ็นเซอร์ไม่ทำงานตามที่โปรแกรมไว้ สิ่งแรกที่พวกเขาจะทำการตรวจสอบ คือ การเรียกใช้โปรแกรม Blink อย่างที่เคยแนะนำวิธีการเปิดโปรแกรม Blink ไปแล้วในตอนแรก ก่อนที่จะทำการเชื่อมต่อพอร์ต USB "COM5" File >>> Examples (ตัวอย่าง) >>> Basics (พื้นฐาน) >>> Blink (การกะพริบ) หรือโปรแกรมสั่งให้ LED (L) กะพริบ

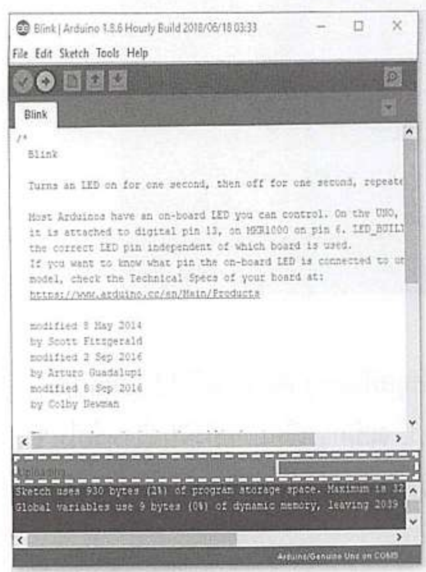
เมื่อเลื่อนลงมาด้านล่างสุด เราจะเห็นโปรแกรมชุดคำสั่งเบื้องต้น หรือ Sketch ของโปรแกรมทดสอบ Blink Sketch หมายถึงชุดคำสั่ง โดยสร้างหรือเขียนขึ้นบนโปรแกรม Arduino IDE และอัพโหลด Sketch นั้น เข้าไปในบอร์ด Arduino เพื่อให้ส่งสัญญาณคำสั่งให้ชุดโมดูล หรือวงจรอิเล็กทรอนิกส์ทำงานตามที่ต้องการ (อธิบายซ้ำ) ซึ่งถูกกำหนดไว้ในโปรแกรม Arduino IDE ไว้เป็นค่าเบื้องต้น (จากโรงงาน)





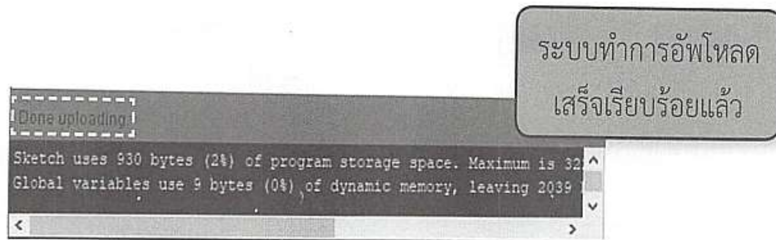
หลังจากที่เชื่อมต่ออุปกรณ์แล้ว หมายถึง เลือกพอร์ตที่ถูกต้องในโปรแกรม Arduino IDE และทำการ Update Driver ในการตั้งค่า Device Manager แล้ว ก็เริ่มจากให้ทำการคลิกที่เครื่องหมายถูก เพื่อทำการ Compile



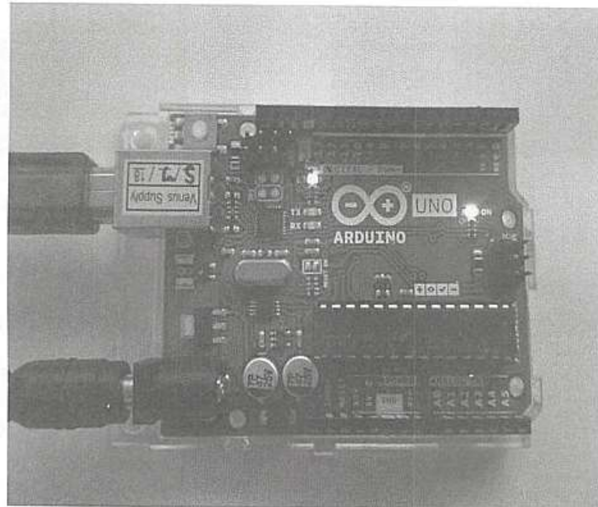


เมื่อระบบทำการ Compile เสร็จแล้ว ให้ทำการคลิกที่ปุ่มลูกศรที่ชี้ไปทางขวามือเรา เพื่อทำการอัพโหลดโปรแกรม Blink ไปยังบอร์ด Arduino

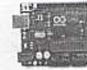
ระบบกำลังทำการอัพโหลดโปรแกรมไปยังบอร์ด Arduino

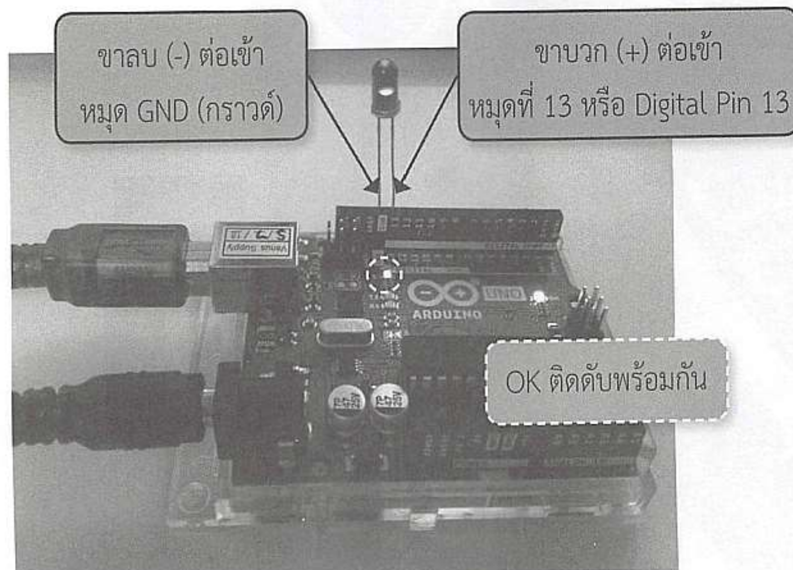


เมื่อโปรแกรมทำการอัพโหลดเสร็จเรียบร้อยแล้ว บอร์ด Arduino จะทำงานทันที โดยไฟ LED L จะติดสว่าง 1 วินาที และดับ 1 วินาที วนซ้ำตามที่ระบบได้เขียนโปรแกรมไว้ และนั่นแสดงว่า บอร์ด Arduino ของเรา พร้อมที่จะสร้างความสนุกแล้ว ด้วยความคิดสร้างสรรค์นับพันล้านโปรเจ็กต์ตามที่เราต้องการ



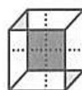
ทีนี้เรามาลองต่ออุปกรณ์ LED ที่หมุดดิจิตอล คือ หมุดที่ 13 และหมุด GND (กราวด์) ที่อยู่ติดกัน ลองดูว่า LED L จะมีการเชื่อมต่อถึงกันจริงหรือไม่ เริ่มจากวิธีแรกเสียบขาอุปกรณ์ LED เชื่อมต่อเข้ากับหมุดโดยตรง

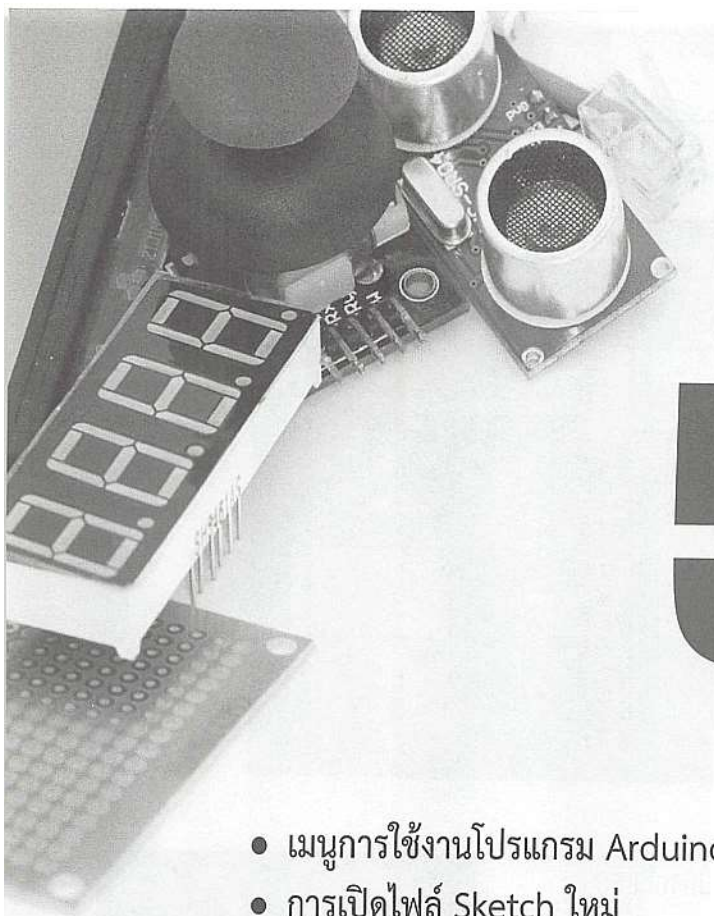




แต่ถ้าเราต่ออุปกรณ์ LED โดยตรงกับ Digital Pin บนบอร์ด Arduino อาจทำให้กระแสไฟฟ้าที่ไหลเกินย้อนกลับเข้ามาทำอันตรายบอร์ดได้ อย่างน้อยอายุการใช้งานของบอร์ดก็จะสั้นลง “ยังไม่สมควรตาย ก็ดันมาตาย” เช่นนั้นเราอาจต่อวงจรบนบอร์ดทดลอง โดยการนำอุปกรณ์ตัวต้านทานต่อแทรกเข้าไปก่อนที่ขาบวก (+) ของอุปกรณ์ LED จะได้รับสัญญาณไฟ







5

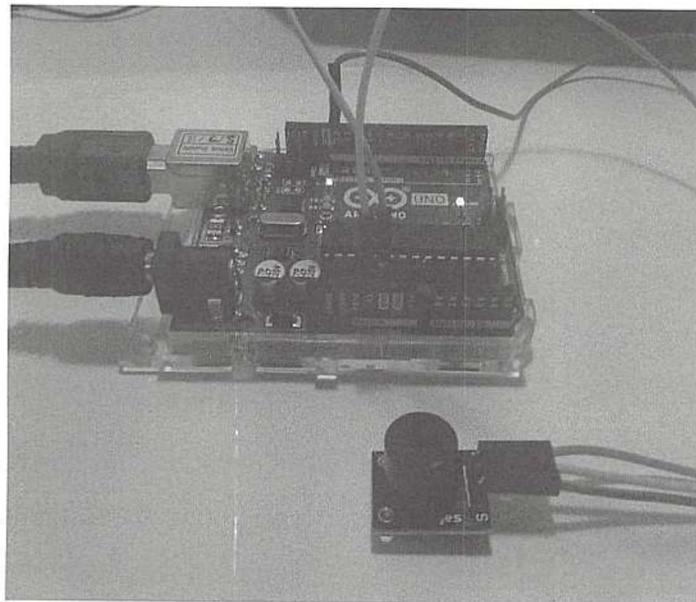
- เมนูการใช้งานโปรแกรม Arduino IDE
- การเปิดไฟล์ Sketch ใหม่
- การต่อวงจร และการสร้างโปรแกรม Sketch

ลงมือเขียนโปรแกรม

หนังสือเล่มนี้เน้นทำให้ผู้อยากจะใช้บอร์ด Arduino ใช้ได้ใช้เป็น เพื่อเป็นการปูพื้นฐาน "ความอยากที่จะเรียนรู้" ให้กับผู้ที่สนใจ อย่างที่เคยบอกไปแล้วในบทแรก การที่คนเราจะเรียนรู้ในสิ่งที่ไม่เคยรู้ หรือไม่เคยมีประสบการณ์ จำเป็นจะต้องเกิดความเข้าใจถึงสิ่งนั้นเป็นอย่างดีก่อน "ความเข้าใจที่ไม่ได้เกิดจากการท่องจำ" เหมือนอย่างการเรียนรู้ภาษา "เริ่มจากเมื่อเราจำความได้" เราต้องฟังการพูด ฟังคำศัพท์ต่างๆ จากพ่อ แม่ พี่ ป้า น้า อา ญาติพี่น้อง จนเราเริ่มจำคำศัพท์ได้ จากนั้นจึงเริ่มหัดพูด เมื่อถึงวัยที่ต้องไปโรงเรียน เราก็เริ่มที่จะหัดเขียน นี่คือแนวทางการเรียนที่ถูกต้อง และสัมฤทธิ์ผลมากที่สุด สำคัญที่สุด หากเป็นการเรียนรู้อะไรที่ยุ่งยาก และมีความซับซ้อน "เช่น ภาษา C หรือการเขียนโปรแกรมบน Arduino IDE" จำเป็นมากที่เราจะต้องมีสิ่งเร้า เพื่อให้เกิดกำลังใจอยากที่จะเรียนรู้ต่อไปเรื่อยๆ

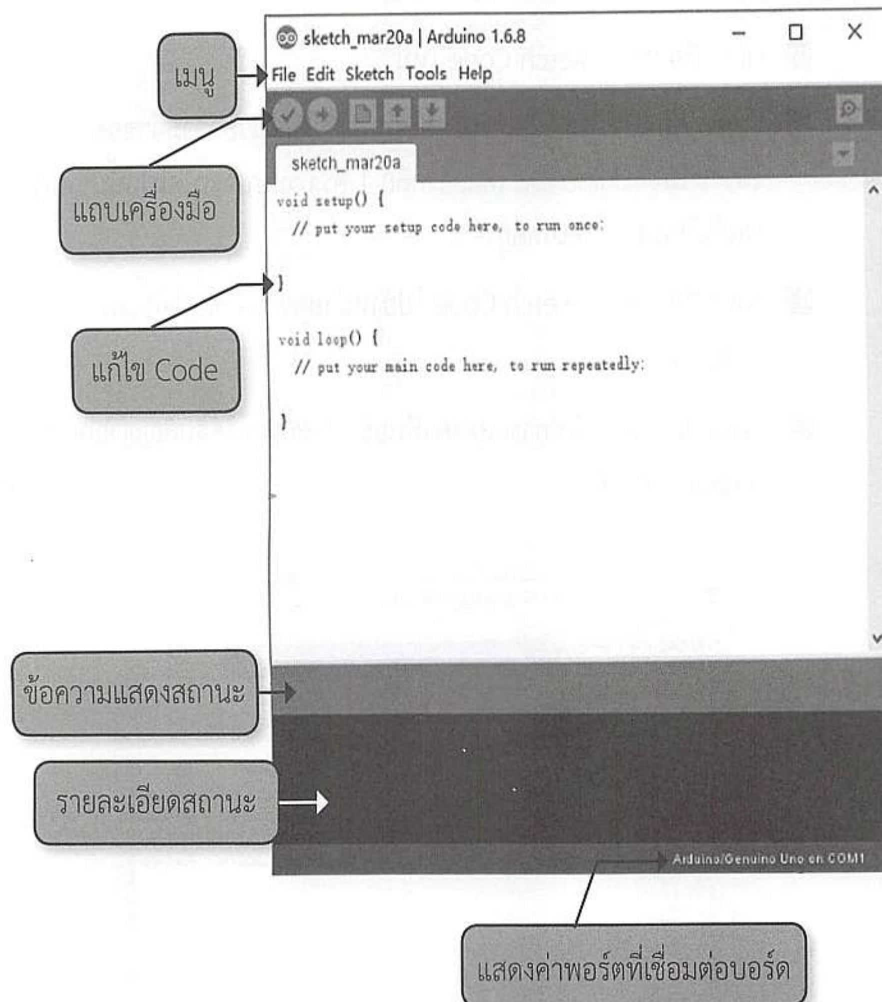
"กำลังใจนั้นเกิดจากอะไร" กำลังใจนั้นเกิดได้จาก "ความสนุก" เมื่อใดที่เราลงมือทำอะไรสักอย่างและเกิดความสนุก "ความมันส์จะเกิดขึ้น" และอยากที่จะทำมันอีก เพราะฉะนั้นการฟัง ตามด้วยการทำความเข้าใจ และการหัดทำตามคนอื่นหรือเอาคนอื่นเป็นแบบอย่าง คือ การสร้างพื้นฐานของประสบการณ์ เพื่อ

การเรียนรู้ที่ดีที่สุด" (เหมือนอย่างเราพยายามเลียนแบบพ่อและแม่ เมื่อเราเริ่มจะมีความเข้าใจอะไรมากขึ้น)

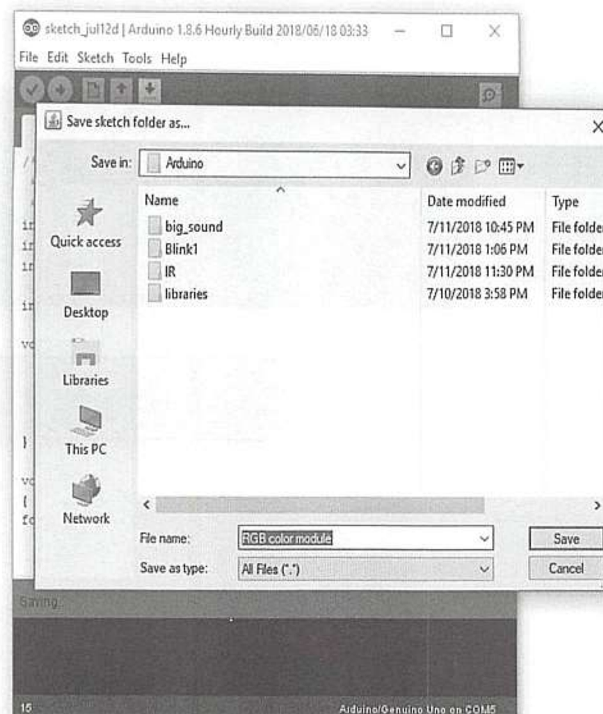


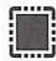
หลายคนเมื่อได้อ่านหนังสือเล่มนี้ คงจะตัดสินใจเปิดข้ามมาบทนี้ "ซึ่งนั่นเป็นสิ่งที่คุณตัดสินใจได้อย่างถูกต้อง" เพราะมันจะไม่ทำให้เรา "เบื่อ" มากไปกว่านี้ อย่าพึ่งไปหมกมุ่นกับเนื้อหาที่เรายังไม่เข้าใจ จงลงมือปฏิบัติเพื่อให้เกิดประสบการณ์กับบอร์ด Arduino โปรแกรม Arduino IDE ลองเชื่อมต่อบอร์ด Arduino กับโปรแกรม Arduino IDE บนคอมพิวเตอร์ ลองเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์ หรือชุดโมดูลอิเล็กทรอนิกส์กับบอร์ด Arduino และลองใช้โปรแกรม Arduino IDE ลอง Copy โปรแกรมชุดคำสั่ง หรือ Sketch ของผู้อื่นหรือผู้รู้ มาวางใน Sketch ของเราเอง บนโปรแกรม Arduino IDE ที่เราสร้างใหม่เพื่อ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino ให้เกิดการทำงานในวงจรอิเล็กทรอนิกส์ จากนั้นก็ค่อยมาพิจารณาถึงเหตุผลว่า "ทำไมต้องเขียนโปรแกรมด้วยคำๆ นี้ ประโยคๆ นี้ วงเล็บนั้น ปีกกาปีกนั้น" แล้วเราจะเข้าใจได้เอง โดยอาจไม่จำเป็นต้องใช้ความพยายามด้วยซ้ำ

เมนูการใช้งานโปรแกรม Arduino IDE



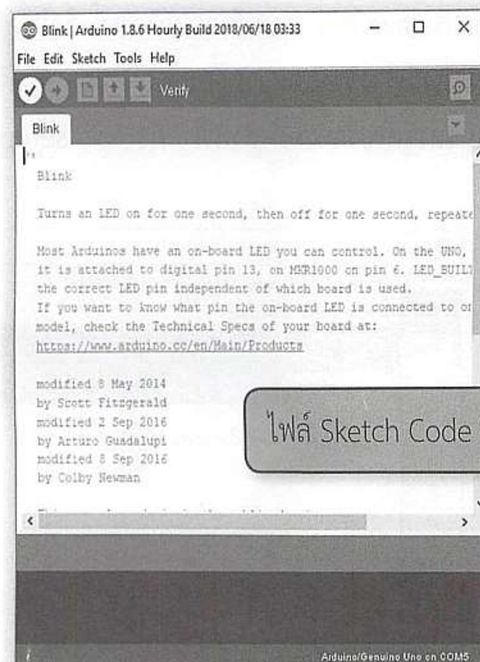
- Verify คือ ตรวจสอบโค้ดของคุณเพื่อรวบรวมข้อผิดพลาด
- Update คือ รวบรวมโค้ดของคุณและอัพโหลดไปยังบอร์ดที่กำหนดค่า
- NEW คือ สร้าง Sketch Code ใหม่
- Open คือ เปิดไฟล์ Sketch Code ที่บันทึกในหน่วยความจำของโปรแกรม Arduino IDE โดยการคลิก 1 ครั้ง ระบบจะเปิดไฟล์นั้นขึ้นมาโดยไม่ปิดไฟล์เก่าที่เปิดอยู่
- Save คือ บันทึก Sketch Code ไปยังหน่วยความจำของโปรแกรม Arduino IDE
- Serial Monitor คือ การแสดงค่าที่บอร์ด Arduino ได้รับสัญญาณจากโมดูลอิเล็กทรอนิกส์



 การเปิดไฟล์ Sketch ใหม่

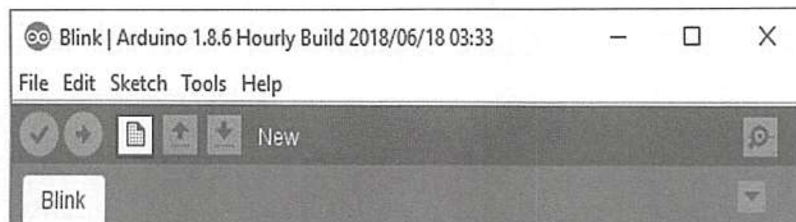
เมื่อเราเปิดโปรแกรม Arduino IDE ขึ้นมาแล้ว โดยปกติระบบจะเกิดไฟล์ Sketch ที่เราเปิดล่าสุดก่อนที่เราจะทำการปิดโปรแกรม "เมื่อวานเปิดไฟล์นี้แล้วปิด เมื่อเปิดโปรแกรมอีกไฟล์นั้นจะถูกเปิดขึ้นอีกครั้ง" ถ้าเรายังไม่ได้ทำการ Save ไฟล์ใหม่ ก็แน่นอนว่า คลัง Sketch ในระบบโปรแกรม Arduino IDE ของเรา จะมีเพียงไฟล์ Sketch ของ Blink ซึ่งเป็น Sketch พื้นฐานที่ใช้ตรวจสอบสถานะความพร้อมใช้งานของบอร์ด Arduino และ Blink จะติดมากับโปรแกรม Arduino IDE หลังจากการติดตั้งโปรแกรม



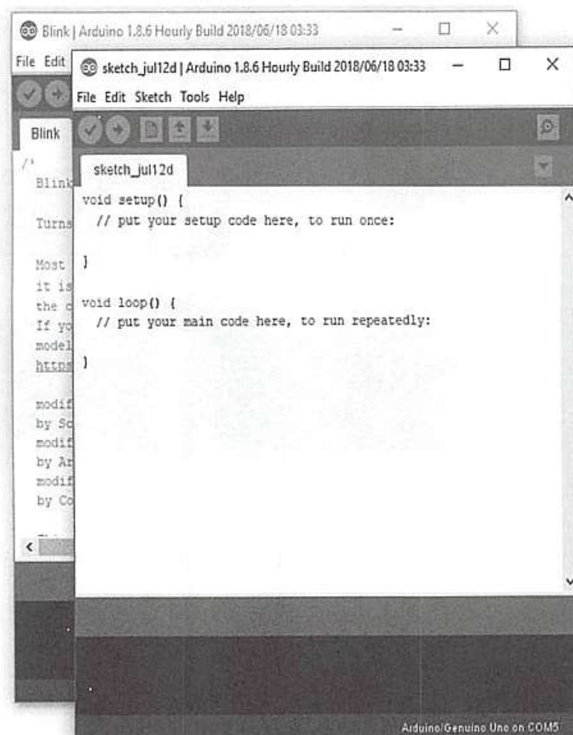


หากเราต้องการสร้างไฟล์ Sketch ใหม่ เพื่อเขียนโปรแกรม และ Upload ไปยังบอร์ด Arduino ซึ่งได้มีการเชื่อมต่อกับวงจรโมดูลอิเล็กทรอนิกส์แล้ว "หรือ

ยังไม่เชื่อมต่อก็ได้ ตามใจเลย" ให้เราไปคลิกเลือกที่ NEW เพื่อสร้างไฟล์ Sketch ใหม่



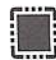
ระบบจะเปิดไฟล์ Sketch ขึ้นมาใหม่ เพื่อชื่อชั่วคราวว่า "Sketch_jul12d" จนกว่าจะมีการ Save ไฟล์ ระบบจึงจะให้มีการตั้งชื่อไฟล์ Sketch ใหม่ โดยไฟล์ Sketch ใหม่ที่ถูกสร้างขึ้นนี้ จะมาพร้อมด้วยตัวอย่างชุดคำสั่งเริ่มต้น ซึ่งเป็นคำสั่งพื้นฐานของชุดโปรแกรมคำสั่งทั่วไป คือ void setup หมายถึง การกำหนด



ค่าหมุดบน Arduino เชื่อมต่อกับขาของอุปกรณ์ในวงจร และ void loop หมายถึง เงื่อนไขการทำงานของอุปกรณ์ในวงจร เช่น วนซ้ำ ติดกี่วินาที ดับกี่วินาที เป็นต้น ซึ่งถ้าเราเข้าใจและสามารถเขียนโปรแกรมด้วยภาษา C ได้แล้ว ก็ลงมือเขียนโปรแกรมชุดคำสั่งได้เลย

อย่างที่บอก "การหัดทำตามคนอื่นหรือเอาคนอื่นเป็นแบบอย่าง คือ การสร้างพื้นฐานของประสบการณ์ เพื่อการเรียนรู้ที่ดีที่สุด" เช่นนั้นเรามาลองเขียนโปรแกรมชุดคำสั่งที่ Copy มาจากผู้รู้ ซึ่งพวกเขาเผยแพร่ในอินเตอร์เน็ต เพื่อนำมาใช้งานในโปรแกรม Arduino IDE สำหรับการควบคุมการทำงานชุดโมดูลอิเล็กทรอนิกส์ ผ่านบอร์ด Arduino ของเรา

โดยตัวอย่างนี้ คือ โปรเจ็กต์ RGB color module ซึ่งคิดว่า เป็นชุดโมดูลที่ง่ายรองลงมาจาก Blink แต่ไม่ถึงกับเด็กๆ จนเกินไป ถ้าเช่นนั้นเรามาลงปฏิบัติไปพร้อมๆ กันเลย

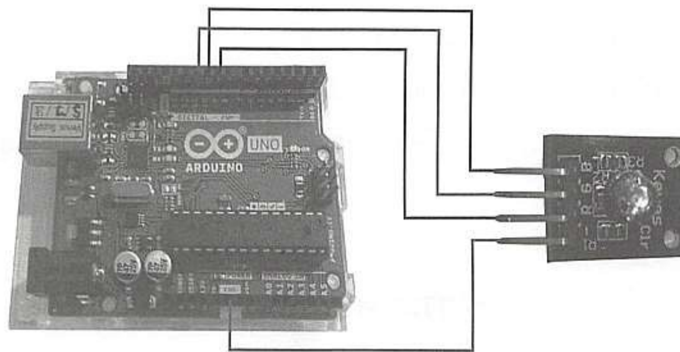
 การต่อวงจร และการสร้างโปรแกรม Sketch "เอาแบบอย่างผู้รู้"

RGB color module คือ วงจรแสดงสี RGB R คือ Red (สีแดง) G คือ Green (สีเขียว) และ B คือ Blue (สีน้ำเงินหรือสีฟ้า) ซึ่งไฟทั้ง 3 สี จะอยู่ในหลอด LED เดียวกัน โดยการเขียนโปรแกรม Sketch จะกำหนดเงื่อนไขให้ไฟทั้ง 3 สีติดสลับกันวนไป

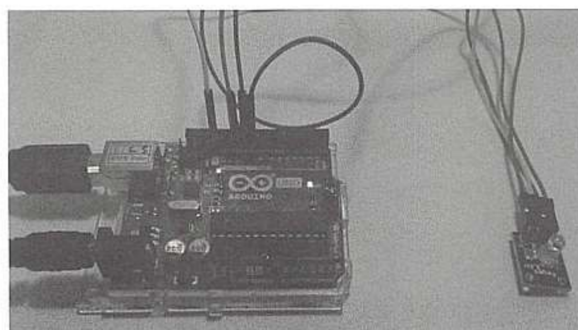
 เริ่มจากต่อวงจร RGB color module

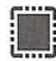
นี่คือชุดวงจร RGB color module โดยขาอุปกรณ์ จะมีอยู่ด้วยกัน 4 ขา คือ (ไล่จากซ้ายของเราไปขวา)

- ขาที่ 1 คือ B
- ขาที่ 2 คือ G
- ขาที่ 3 คือ R
- ขาที่ 4 คือ - หรือ GND (กราวด์)

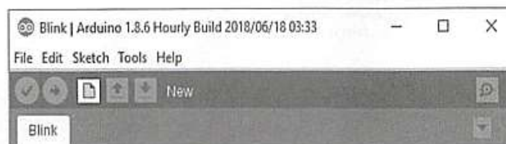


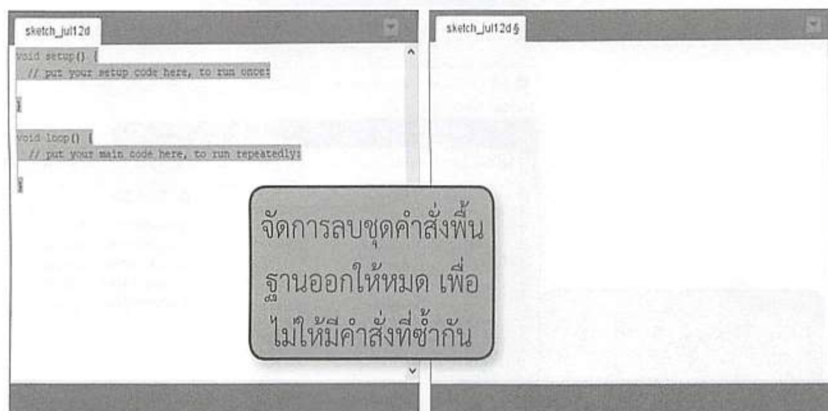
1. ขา B เชื่อมต่อกับหมุด -10 บอร์ด Arduino
2. ขา G เชื่อมต่อกับหมุด -11 บอร์ด Arduino
3. ขา R เชื่อมต่อกับหมุด -9 บอร์ด Arduino
4. ขา - (GND) เชื่อมต่อกับหมุด GND บอร์ด Arduino



 การสร้างโปรแกรม Sketch RGB color module

เมื่อต่อวงจรเรียบร้อยแล้ว ให้ทำการเขียนโปรแกรม Sketch โดยการ Copy จากผู้รู้ เพื่อสั่งสมประสบการณ์ที่มีต่อบอร์ด Arduino โดยเริ่มจากสร้างไฟล์ Sketch ขึ้นมาใหม่ (ขั้นตอนเดิมการสร้างไฟล์ใหม่)





จากนั้น Copy ชุดคำสั่ง sketch code ของผู้รู้มาวางไว้ในไฟล์ Sketch RGB color module ของเรา

```
/* RGB module project code
 * tutorial url: http://osoyoo.com/?p=549
 */
int redpin = 11; //select the pin for the red LED
int bluepin =10; // select the pin for the  blue LED
int greenpin =9;// select the pin for the green LED

int val;

void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
for(val=255; val>0; val--)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
for(val=0; val<255; val++)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
 Serial.println(val, DEC);
}
```

จากนั้นนำไปวางไว้ในไฟล์ Sketch RGB color module ในโปรแกรม Arduino IDE ของเรา เช่นเดียวกันสำหรับวิธีการวาง ไม่ว่าจะ Paste หรือ Ctrl + V ก็แล้วแต่ตามใจ

แล้วทำการ Save ไฟล์ Sketch RGB color module เพื่อบันทึกเข้าไปในระบบโปรแกรม Arduino IDE ให้เรียบร้อย "กันเหนียว"

Save

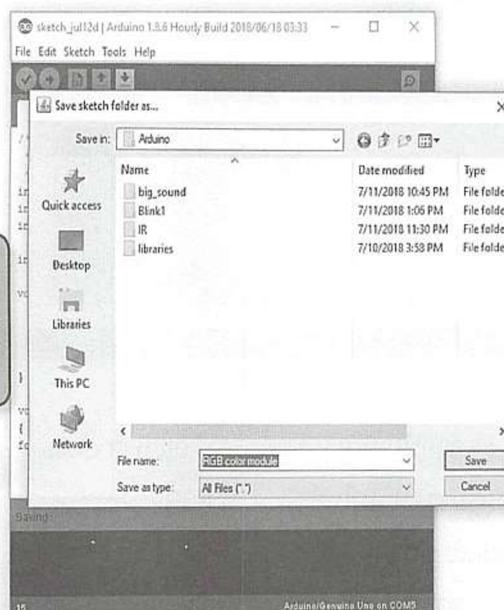
sketch_jul12d §

```
/* RGB module project code
 * tutorial url: https://osoyoo.com/?p=548
 */
int redpin = 11; //select the pin for the red LED
int bluepin =10; // select the pin for the  blue LED
int greenpin =9;// select the pin for the green LED

int val;

void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
for(val=255; val>0; val--)
  {
   analogWrite(11, val);
```



ตั้งชื่อไฟล์ เช่น RGB color module แล้วคลิกที่ Save เป็นอันเสร็จพิธี

Done Saving

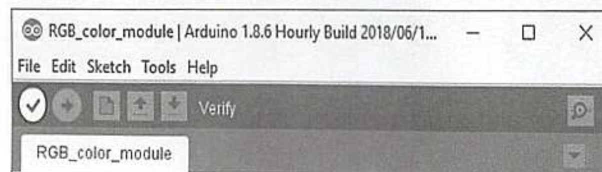
The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

15 Arduino/Genuino

ระบบทำการ Save เรียบร้อยแล้ว

 การ Verify หรือ Compile และ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino

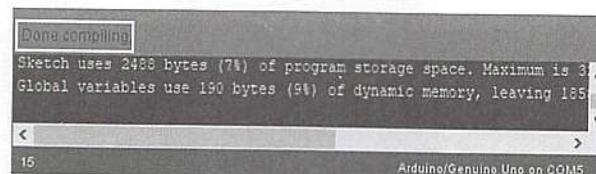
เมื่อ Sketch พร้อมแล้วให้เราทำการ Verify หรือ Compile เพื่อทำการตรวจสอบและยืนยันชุดคำสั่งใน Sketch ก่อนที่จะทำการ Upload Sketch ไปยังบอร์ด Arduino ที่มีการเชื่อมต่อกับชุด RGB color module แล้ว โดยคลิกที่เครื่องหมายถูก



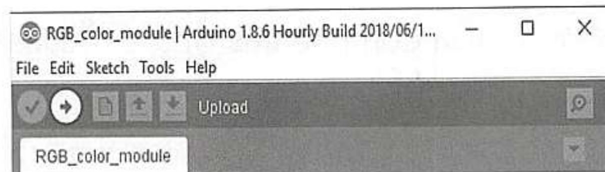
ระบบกำลังทำการ Compile Sketch



ระบบ Compile Sketch (ตรวจสอบเสร็จแล้ว)



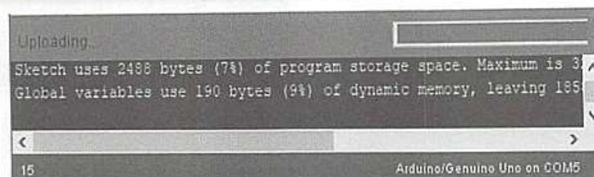
จากนั้นเราก็ทำการ Upload โปรแกรมไปยังบอร์ด Arduino โดยคลิกที่ลูกศรที่หันไปทางด้านขวา



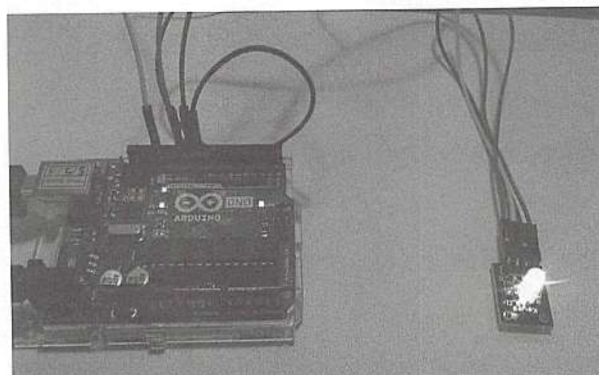
ระบบกำลังทำการ Upload



ระบบทำการ Upload เรียบร้อยแล้ว



เมื่อระบบทำการ Upload แล้ว บอร์ด Arduino จะสตาร์ทการทำงานตามโปรแกรมชุดคำสั่งทันที ชุด RGB color module จะถูกป้อนไฟเข้าวงจรและรับสัญญาณคำสั่งดิจิตอลจากบอร์ด Arduino และทำงานตามโปรแกรมทันที



เมื่อเอาแบบอย่างจากของคนอื่นจนสำเร็จแล้ว ถ้าเราไม่ยอมทำความเข้าใจกับชุดคำสั่ง Sketch Code ต่อไปเราก็ต้อง Copy ของคนอื่นมาใช้อีกเรื่อยไปจนสุดท้าย "ไม่รู้เรื่องอะไรเลย" ต้องยืมจมูกคนอื่นหายใจตลอด เช่นนั้นเราลองไปพิจารณากับแบบเบื้องต้น ถึงชุดคำสั่ง Sketch Code RGB color module "อะไรเป็นอะไร"

sketch_jul12d §

```
/* RGB module project code
 * tutorial url: http://osoyoo.com/?p=549
 */
int redpin = 11; //select the pin for the red LED
int bluepin =10; // select the pin for the  blue LED
int greenpin =9;// select the pin for the green LED

int val;

void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  Serial.begin(9600);
}

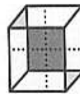
void loop()
{
for(val=255; val>0; val--)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
for(val=0; val<255; val++)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
 Serial.println(val, DEC);
}
```

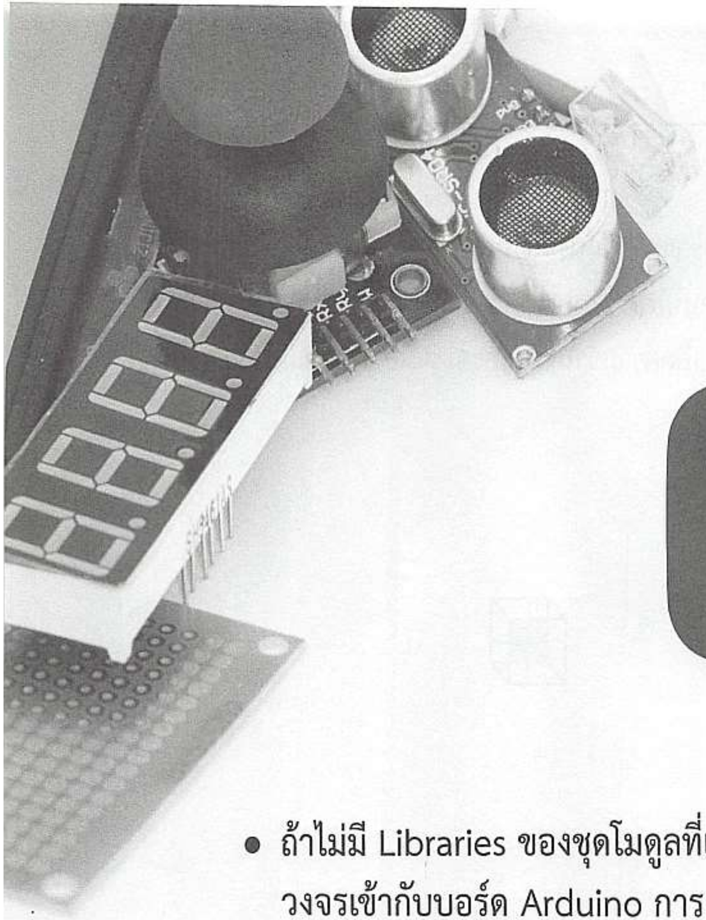
1. Header คำประกาศกำหนดเลือกหมุด Arduino

   Int คือ ตัวแปรซึ่งหมายถึงชื่อเรียกแทนพื้นที่เก็บข้อมูล

   - redpin = 11 หมายถึง กำหนดให้หมุดที่ 11 คือ LED สีแดง
   - bluepin = 10 หมายถึง กำหนดให้หมุดที่ 10 คือ LED สีน้ำเงิน
   - greenpin = 9 หมายถึง กำหนดให้หมุดที่ 9 คือ LED สีเขียว
     - select the pin for the red LED คือ เลือกหมุดนี้สำหรับ LED สีแดง
     - select the pin for the blue LED คือ เลือกหมุดนี้สำหรับ LED สีน้ำเงิน
     - select the pin for the green LED คือ เลือกหมุดนี้สำหรับ LED สีเขียว
2. คำสั่งติดตั้งโปรแกรม ซึ่งมีคำสั่งย่อยดังนี้
3. pinMode คือ วิธีการทำงานหรือการตอบสนองของหมุด และในวงเล็บ ()
   - (redpin, OUTPUT) คือ หมุดสีแดง ส่งสัญญาณออกไป
   - (bluepin, OUTPUT) คือ หมุดน้ำเงิน ส่งสัญญาณออกไป
   - (greenpin, OUTPUT) คือ หมุดสีเขียว ส่งสัญญาณออกไป
4. คำสั่งให้ทำงานตามโปรแกรม ตามข้อ 2 แบบวนซ้ำ ซึ่งมีคำสั่งย่อยดังนี้
5. การกำหนดค่าระดับความสว่างของไฟ LED
6. การเขียนค่าอะนาล็อก ซึ่งทำให้เกิดแสงไฟ LED สว่างในหลายระดับ 128-255
7. หน่วงเวลาการติด-ดับ 1 วินาที
8. การกำหนดค่าระดับความสว่างของไฟ LED
9. การเขียนค่าอะนาล็อก ซึ่งทำให้เกิดแสงไฟ LED สว่างในหลายระดับ 128-255
10. หน่วงเวลาการติด-ดับ 1 วินาที

นี่คือ การเรียนรู้การเขียนโปรแกรมในภาษา C แบบเบื้องต้น แน่นอนถ้าเรายังสนุกกับการใช้งานบอร์ด Arduino เราจะเรียนรู้และสั่งสมประสบการณ์ทั้งหมดนี้ไปเรื่อยๆ จนเกิดความชำนาญ และจะไม่มีอะไรในโลกนี้ ที่ยากเกินความสามารถของเรา





6

- ถ้าไม่มี Libraries ของชุดโมดูลที่เราจะเชื่อมต่อวงจรเข้ากับบอร์ด Arduino การ Compile หรือ Upload โปรแกรมไปยังบอร์ด Arduino จะไม่สามารถเกิดขึ้นได้
- เมื่อทำการเพิ่ม Libraries ลงไปในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE แล้ว ระบบจะให้เราทำการ Update ทันที โดยให้สังเกตที่ใต้ช่อง "รายละเอียดสถานะ"

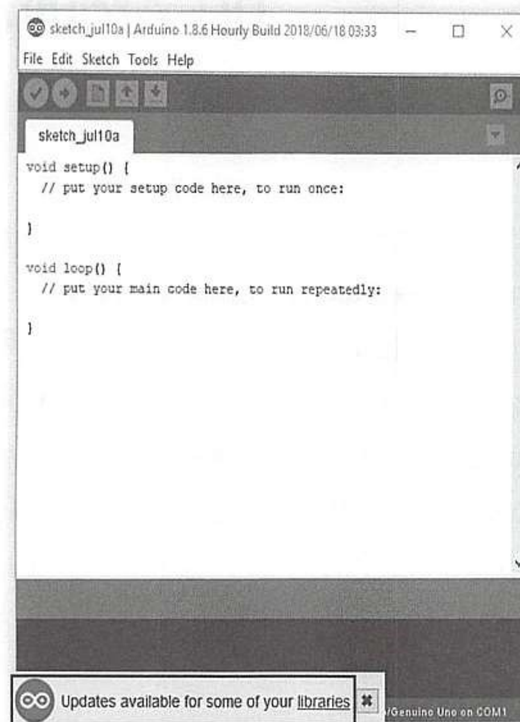
การติดตั้ง Library บน Arduino IDE

Libraries ถ้าจะทำความเข้าใจแบบง่ายๆ ก็คือ โค้ดเฉพาะของชุดโมดูลในแต่ละชุด ซึ่งผู้สร้างหรือผู้ผลิตชุดโมดูลต่างๆ จะต้องสร้าง Libraries ขึ้นมา เพื่อช่วยเหลือการสร้างหรือเขียนโปรแกรมบน Arduino IDE เช่น สมมุติว่า เราจะต่อวงจร Infrared Receiver Module (กดรีโมต LED ติด กดรีโมตอีกครั้ง LED ดับ) โดยเขียนโปรแกรมบน Arduino IDE เพื่อให้บอร์ด Arduino ควบคุมการทำงานของวงจรโมดูล Infrared เราก็จำเป็นต้องหา Libraries ของวงจร Infrared Receiver Module มาบันทึกไว้ใน Libraries ของโปรแกรม Arduino IDE เพื่อให้ระบบโปรแกรมสามารถทำการสนับสนุน Sketch ของโปรแกรม Infrared Receiver Module ได้ "เข้าใจง่ายมั้ย ^^!"

ถ้ายังไม่เข้าใจ ก็เอาเป็นว่า ถ้าไม่มี Libraries ของชุดโมดูลที่เราจะเชื่อมต่อวงจรเข้ากับบอร์ด Arduino การ Compile หรือ Upload โปรแกรมไปยังบอร์ด Arduino จะไม่สามารถเกิดขึ้นได้

โดยปกติโปรแกรม Arduino IDE จะมี Libraries ของมันเองอยู่แล้ว ซึ่งก็ครอบคลุมอยู่พอสมควร อีกทั้งระบบ Arduino IDE ยังให้เราสามารถกด Update Libraries อยู่เรื่อยๆ เพื่อเพิ่ม Libraries ตัวใหม่เข้าไปในโปรแกรม เมื่อผู้พัฒนาทั่วโลกสร้าง Libraries ตัวใหม่ขึ้น "นี่คือข้อดีของการเป็น Open-Source"

ไม่มีลิขสิทธิ์ เซียน Arduino ทั่วทั้งโลก จึงสามารถพัฒนาซอฟต์แวร์ และมีการสนับสนุนให้เกิดความร่วมมือกันอย่างเสรี สำหรับในบริบทของ Libraries จึงทำให้ Libraries มากมายเกิดขึ้น และไม่ว่าใครก็สามารถนำเอาไปใช้ได้ แม้จะเป็นการนำไปใช้ในเชิงธุรกิจก็ตาม เช่นนั้นทุกคนสามารถนำเอาไปประยุกต์ปรับใช้ หรือก๊อบปี้เอาไปประดิษฐ์เครื่องช็อตยุงขายเป็นธุรกิจก็ได้ ไม่มีใครห้าม และที่สำคัญ "มันถูกกฎหมายด้วย"



แต่อย่าลืมว่าโปรเจ็กต์ที่มีบอร์ด Arduino ควบคุมการทำงานนั้น สามารถเกิดขึ้นได้ตลอดเวลา หรือชุดโมดูลบางรุ่น บางยี่ห้อ หรือต่างผู้ผลิตกัน ย่อมจะต้อง Libraries ที่แตกต่างกัน "เรื่องนั้นเราอาจไม่ต้องใส่ใจ" สนใจแค่ว่า ชุดโมดูล

ที่เราจะเอามาเชื่อมต่อกับบอร์ด Arduino โดยเขียนโปรแกรมบน Arduino IDE เพื่อให้บอร์ด Arduino ควบคุมการทำงานของชุดโมดูล ในโปรแกรม Arduino IDE "มี Libraries ของชุดโมดูลที่เราจะใช้หรือไม่" ใส่ใจแค่นี้ก็คงพอ

Name	Date modified	Type	Size
drivers	6/18/2018 3:33 PM	File folder	
examples	6/18/2018 3:33 PM	File folder	
hardware	6/18/2018 3:33 PM	File folder	
java	6/18/2018 3:33 PM	File folder	
lib	6/18/2018 3:33 PM	File folder	
libraries	7/12/2018 12:16 AM	File folder	
reference	6/18/2018 3:33 PM	File folder	
tools	6/18/2018 3:33 PM	File folder	
tools-builder	6/18/2018 3:33 PM	File folder	
arduino	6/18/2018 3:33 PM	Application	395 KB
arduino.l4j	6/18/2018 3:33 PM	Configuration settings	1 KB
arduino_debug	6/18/2018 3:33 PM	Application	393 KB
arduino_debug.l4j	6/18/2018 3:33 PM	Configuration settings	1 KB
arduino-builder	6/18/2018 3:33 PM	Application	11,733 KB
libusb0.dll	6/18/2018 3:33 PM	Application extension	43 KB
msvcp100.dll	6/18/2018 3:33 PM	Application extension	412 KB
msvcr100.dll	6/18/2018 3:33 PM	Application extension	753 KB
revisions	6/18/2018 3:33 PM	Text Document	85 KB
wrapper-manifest	6/18/2018 3:33 PM	XML Document	1 KB

ถ้าเช่นนั้นสมมุติว่า เราจะประดิษฐ์รังสรรค์ผลงานชิ้นเอกทางอิเล็กทรอนิกส์สักชุดหนึ่ง โดยมีบอร์ด Arduino เป็นพระเอกของโปรเจ็กต์ ก่อนอื่นเราจำเป็นจะต้องตรวจสอบว่า โปรแกรม Arduino IDE มี Libraries ของชุดโมดูลอิเล็กทรอนิกส์ที่เรากำลังจะสร้างขึ้นหรือไม่ เช่น สมมุติว่าเราจะประดิษฐ์ชุด Touch sensor moduel เพื่อเป็นอุปกรณ์เซ็นเซอร์ในการตรวจจับการใช้กุญแจผีของโจรขโมย "ที่มีอยู่เต็มบ้านเต็มเมือง" เมื่อใดก็ตามที่มีคนเอาโลหะมาแตะที่รูกุญแจประตูเข้าบ้าน ในยามที่เราไม่อยู่บ้าน หรือในเวลาที่เรานอนหลับ ไฟฉุกเฉินจะติดสว่างขึ้นทันที "ยังกะไฟไซเรนรถตำรวจปราบปรามเลยทีเดียว"

นี่อาจเป็นโปรเจ็กต์มูลค่าพันล้านที่อาจโด่งดังในอนาคต แต่มันจะไม่มีทางสำเร็จได้เลย ถ้าโปรแกรม Arduino IDE ไม่มี Libraries ของชุดเซ็นเซอร์ Touch sensor moduel เช่นนั้นก่อนอื่น ไม่ว่าเราจะเชื่อมต่อชุดโมดูลใดกับบอร์ด Arduino เราจะต้องเข้าไปตรวจสอบในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE ว่า มี Libraries ของชุดโมดูลที่เราจะใช้ทำงานหรือไม่ โดยเข้าไปที่โฟลเดอร์ซึ่งเก็บโปรแกรม Arduino IDE แล้วเข้าไปที่โฟลเดอร์ Libraries

จากนั้นลองตรวจสอบดูว่า มี Libraries ของชุดโมดูลที่เราจะทำการเชื่อมต่อรับคำสั่งจากบอร์ด Arduino หรือไม่ เช่น ตามตัวอย่างเดิมเราจะต่อวงจร Infrared Receiver Module (กดรีโมต LED ติด กดรีโมตอีกครั้ง LED ดับ) ให้ลองตรวจสอบดูว่า มี Libraries ของชุดโมดูลนี้หรือไม่ แน่นอน Infrared Receiver อาจจะต้องขึ้นต้นชื่อด้วย IR

Name	Date modified	Type
Adafruit_Circuit_Playground	7/18/2017 11:21 PM	File folder
Bridge	8/25/2016 5:20 PM	File folder
Esplora	5/20/2015 5:10 PM	File folder
Ethernet	6/18/2018 3:33 PM	File folder
Firmata	3/19/2017 12:47 AM	File folder
GSM	6/18/2018 3:33 PM	File folder
Keyboard	3/8/2016 5:05 PM	File folder
LiquidCrystal	8/10/2017 10:26 AM	File folder
Mouse	3/8/2016 5:04 PM	File folder
Robot_Control	8/18/2017 11:29 AM	File folder
Robot_Motor	8/18/2017 12:11 PM	File folder
RobotIRremote	8/18/2017 11:25 AM	File folder
SD	1/5/2017 6:00 PM	File folder
Servo	3/20/20	
SpacebrewYun	8/25/20	
Stepper	6/18/20	
Temboo	3/13/20	
TFT	6/18/20	
WiFi	6/18/2018 3:33 PM	File folder

บ่อจี้ไม่มีเลย นี่คือ Libraries ทั้งหมดที่เป็น Libraries พื้นฐานที่ติดมากับโปรแกรม Arduino IDE

ถ้าเรา Compile หรือ Upload โปรแกรมไปยังบอร์ด Arduino ระบบจะแจ้ง Error ทันทีว่า IRremote.h: No such file or directory แปลว่า IRremote.h: ไม่มีไฟล์หรือไดเรกทอรีดังกล่าว ซึ่งการทำงานตามโปรแกรมที่สร้างขึ้น จะไม่มีทางเกิดขึ้นได้เลย

IR | Arduino 1.8.6 Hourly Build 2018/06/18 03:33

File Edit Sketch Tools Help

Verify

IR

```
#include <IRremote.h> //adds the library code to the sketch
const int irReceiverPin = 8; //receiver module S pin is connected
const int ledPin = 7;
IRrecv irrecv(irReceiverPin);
decode_results decodedSignal;

void setup()
{
   Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    irrecv.enableIRIn();
}
boolean lightState = false;
unsigned long last = millis();

void loop()
{
    if (irrecv.decode(&decodedSignal) == true) //this is true if a
    {
```

IRremote.h: No such file or directory

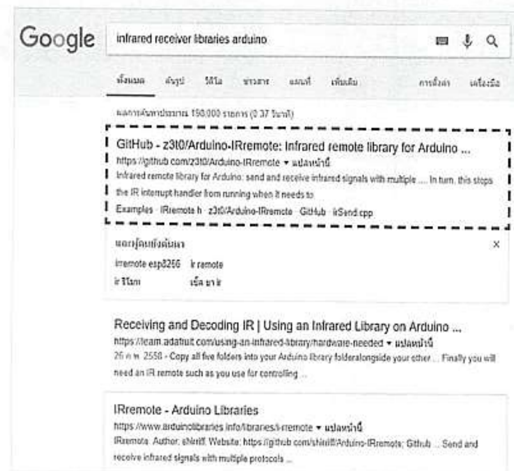
Copy error messages

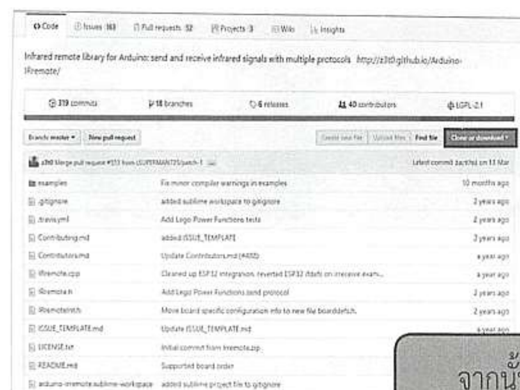
```
exit status 1
IRremote.h: No such file or directory
```

Arduino/Genuino Uno on COM5

เช่นนั้นถ้าไม่มี Libraries ที่ต้องการจะทำยังไง "ขอจากพ่อค้าแถวตลาดบ้านหม้อก็คงไม่ได้" ดังนั้นเราจำเป็นจะต้องพึ่งอากู๋ โดยใช้คีย์เวิร์ด เช่น infrared receiver libraries arduino จากนั้นคลิกที่ค้นหา

เราจะพบผลการค้นหากว่าแสนรายการ ซึ่งอาจเป็น 10 อันดับของหน้าแรกเท่านั้น ที่ตรงกับความต้องการของเรา ซึ่ง Libraries ของ infrared receiver อาจมีหลากหลายรูปแบบ เราจำเป็นจะต้องลองใช้ดูว่า Libraries ตัวไหนเวิร์คที่สุดกับชุดโมดูลของเรา “แต่โดยปกติเวิร์คทุกตัว ต่างกันแค่ Libraries ธรรมดากับ Libraries พิเศษ หมายถึง รองรับ Sketch ได้ในระดับยอดเยี่ยม” ในส่วนของการดาวน์โหลดก็ง่ายๆ สมมุติว่าเราเลือก Link แรก Github



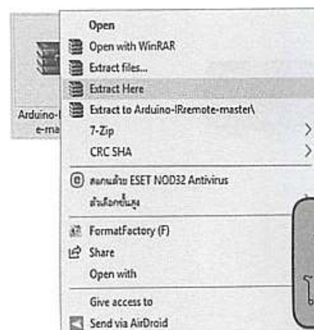


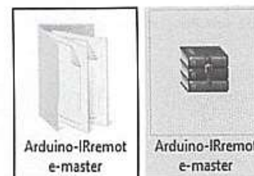
จากนั้นคลิกที่
Clone or download



คลิกที่ Download ZIP

เมื่อไฟล์ ZIP หรือ RAR ถูกดาวน์โหลดมาเก็บไว้ที่หน่วยความจำของคอมพิวเตอร์แล้ว ก็ทำการแตกไฟล์ได้เลย



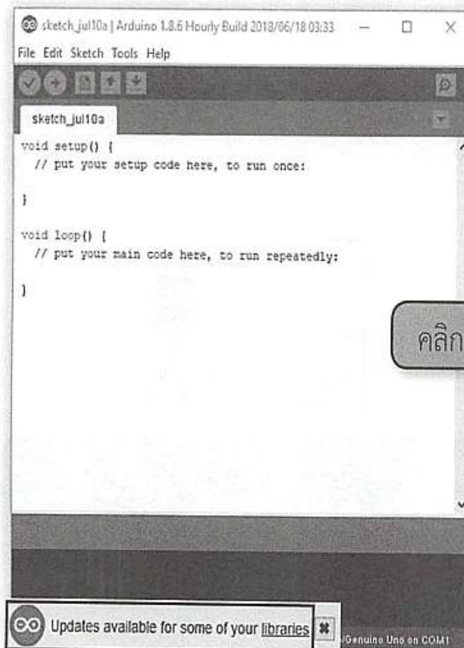


จากนั้น Copy โฟลเดอร์ Libraries ไปเก็บไว้ในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

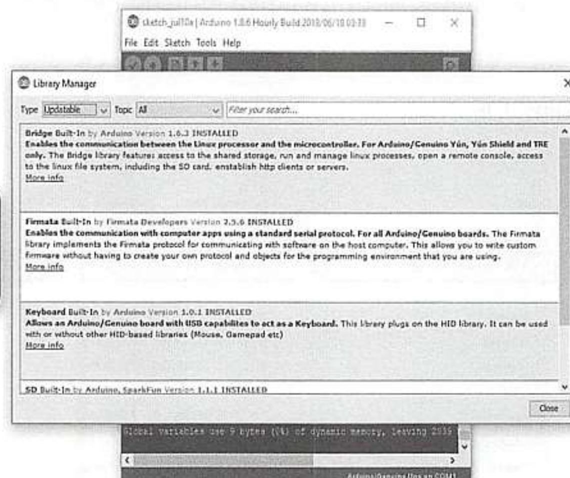
ไปที่โฟลเดอร์ซึ่งเก็บโปรแกรม Arduino IDE แล้วเข้าไปที่โฟลเดอร์ Libraries จากนั้น Paste หรือวางลงในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

Name	Date modified	Type	Size
Adafruit_Circuit_Playground	7/18/2017 11:21 PM	File folder	
Arduino-IRremote-master	7/12/2018 1:36 AM	File folder	
Bridge	8/25/2016 5:20 PM	File folder	
Esplora	5/20/2015 5:10 PM	File folder	
Ethernet	6/18/2018 3:33 PM	File folder	
Firmata	3/19/2017 12:47 AM	File folder	
GSM	6/18/2018 3:33 PM	File folder	
Keyboard	3/8/2016 5:05 PM	File folder	
LiquidCrystal	8/10/2017 10:26 AM	File folder	
Mouse	3/8/2016 5:04 PM	File folder	

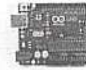
เมื่อทำการเพิ่ม Libraries ลงไปในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE แล้ว ระบบจะให้เราทำการ Update ทันที โดยให้สังเกตที่ใต้ช่อง "รายละเอียดสถานะ"



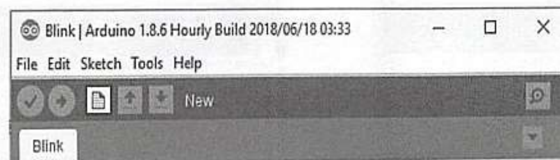
คลิกที่ Libraries



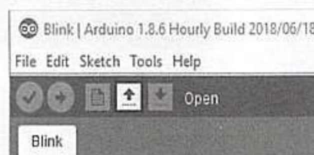
ระบบกำลังทำการ Libraries

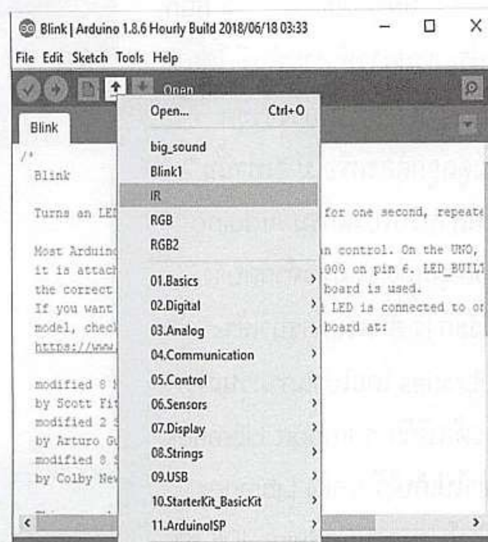


จากนั้นเปิดโปรแกรม Arduino IDE ขึ้นมาใหม่ หลังจากที่เชื่อมต่อชุดโมดูลกับบอร์ด Arduino เรียบร้อยแล้ว ให้เลือกที่ NEW เพื่อทำการเขียนโปรแกรมคำสั่ง

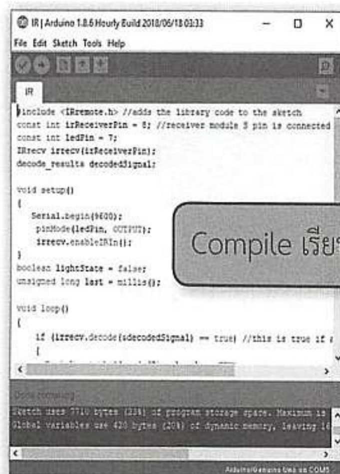


หรือถ้ามีไฟล์โปรแกรมอยู่แล้ว หมายถึง ได้ทำการเขียนโปรแกรม "Sketch" Infrared Receiver Module เอาไว้แล้ว แต่ติดปัญหาโปรแกรมแจ้ง IRremote.h: No such file or directory "ก็ไม่มี Libraries ของโมดูลนั่นแหละ" ก็ให้ทำการเลือกที่ Open จากนั้นเลือกไฟล์ Sketch ของชุดโมดูล เพื่อเปิดขึ้นมาทำงานอีกครั้งได้เลย



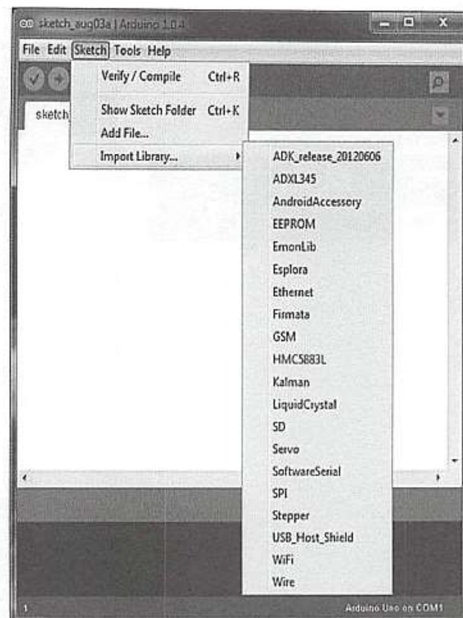


จากนั้นจัดการ Verify หรือ Compile และ Upload โปรแกรม หรือ Sketch ไปยังบอร์ด Arduino ได้เลย





บอร์ด Arduino จะ Run โปรแกรมการทำงานทันที โดยขึ้นอยู่กับว่า เราต่อวงจรชุดโมดูลถูกต้องหรือไม่ "เท่านั้น" แต่สำหรับโปรแกรม Arduino IDE รุ่นเก่า จะมีข้อจำกัดบางประการ สำหรับการนำเข้า Libraries โดยโปรแกรมรุ่นเก่าจะต้องใช้การ Import Libraries เข้าไปเก็บไว้ในคลัง Libraries ของโปรแกรม Arduino IDE ซึ่งบางครั้งอาจมี Libraries บาง

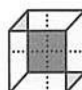


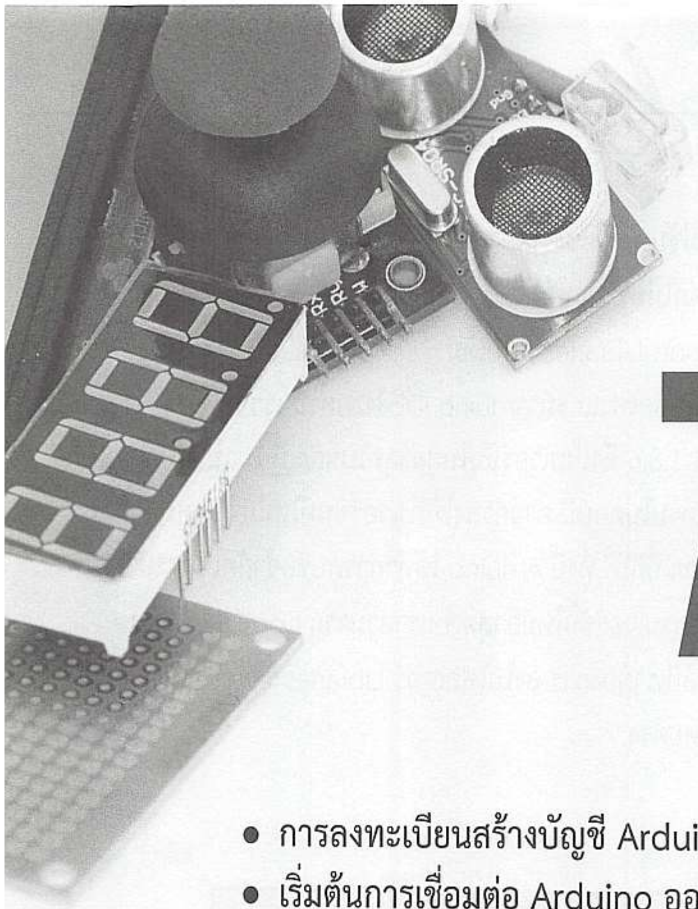
ตัว ที่โปรแกรมแกรมไม่ซัพพอร์ต ซึ่งรุ่นเก่าอาจเป็นเวอร์ชัน 1.0.4 ไปจนถึงรุ่นก่อนเวอร์ชัน 1.8.6 แต่ก็ไม่ทราบแน่ชัดว่า โปรแกรม Arduino IDE เวอร์ชันใดได้ทำการยกเลิกการ Import Libraries แบบเดิม

แต่ถ้าเราดาวน์โหลดโปรแกรม Arduino IDE ในเวลานี้ เราจะได้ Arduino IDE เวอร์ชัน 1.8.6 ซึ่งเป็นเวอร์ชันใหม่ล่าสุดในขณะนี้ ไม่แน่ถ้าใครกำลังดาวน์โหลดโปรแกรมในตอนนี้ อาจกำลังได้รับเวอร์ชันที่ใหม่กว่า และเจ๋งกว่าเวอร์ชันล่าสุดในขณะนี้ก็ได้ ทั้งนี้ Arduino ได้ทำการลบข้อจำกัดในการนำเข้า Libraries ในโปรแกรมเวอร์ชันใหม่ล่าสุด ซึ่งเราสามารถ Copy และ Paste Libraries ของชุดโมดูลที่เราต้องการ ลงในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE ได้เลยโดยตรง

Name	Date modified	Type	Size
Adafruit_Circuit_Playground	7/18/2017 11:21 PM	File folder	
Arduino-IRremote-master	7/12/2018 1:36 AM	File folder	
Bridge	8/25/2016 5:20 PM	File folder	
Esplora	5/20/2015 5:10 PM	File folder	
Ethernet	6/18/2018 3:33 PM	File folder	
Firmata	3/19/2017 12:47 AM	File folder	
GSM	6/18/2018 3:33 PM	File folder	
Keyboard	3/8/2016 5:05 PM	File folder	
LiquidCrystal	8/10/2017 10:26 AM	File folder	
Mouse	3/8/2016 5:04 PM	File folder	

จากนั้นปิดโปรแกรม Arduino IDE แล้วเปิดใหม่ ระบบจะทำการค้นหา Libraries ได้โดยอัตโนมัติ ถึงขั้นตอนนี้เราก็เตรียมตัวสนุกกับบอร์ด Arduino ได้เลย Enjoy...





7

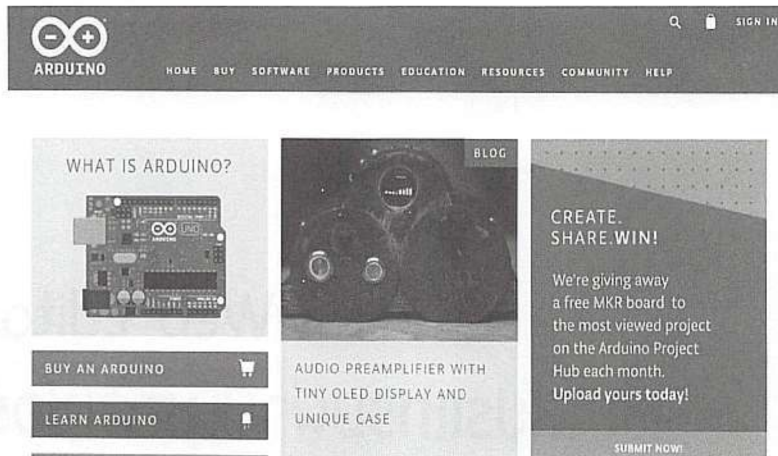
- การลงทะเบียนสร้างบัญชี Arduino
- เริ่มต้นการเชื่อมต่อ Arduino ออนไลน์
- การสร้างไฟล์ Sketch ใหม่ และการนำเข้าไฟล์ Sketch โดยช่องทางปกติ
- การ Verify หรือ Compile และ Upload ไปยังบอร์ด Arduino แบบออนไลน์
- การนำเข้า Libraries ของเว็บไซต์ Arduino Web Editor

Arduino Web Editor เขียนโปรแกรมผ่านบราวเซอร์

บางทีสำหรับเรื่อง Arduino เราอาจกลายเป็นคนไร้เพื่อน เพราะอาจไม่มีใครพูดคุยด้วย หรือไม่มีใครที่พอจะให้เราถามถึงข้อสงสัยหลายอย่าง สำหรับการใช้งานบอร์ด Arduino หรือบางครั้งอาจถามได้ แต่ก็ไม่มีใครพอจะตอบคำถามของเราได้ โดยเฉพาะปัญหาที่เกิดขึ้นกับการใช้ของบอร์ด Arduino ซึ่งจำเป็นจะต้องถาม "ผู้รู้" ในเรื่องของ Arduino จึงจะสามารถได้คำตอบที่ Complete และแน่นอนปัญหาไร้เพื่อนแบบนี้เกิดขึ้นทั่วทุกมุมโลก เพราะบอร์ด Arduino เป็นที่นิยมสำหรับผู้คนทั่วโลก

อีกทั้งระบบปฏิบัติการของ Arduino คือ Code ภาษา C ที่เป็น Open-Source จึงทำให้ผู้คนหลากหลายสาขาสามารถนำไปพัฒนาต่อยอดได้แบบไร้ขีดจำกัด เริ่มตั้งแต่คนที่มีความสนใจ Arduino เพียงเป็นอดิเรก นักศึกษา โปรแกรมเมอร์ ไปจนถึงวิศวกร จนในที่สุดกลายเป็นความร่วมมือพัฒนา Arduino แบบเปิดกว้างอย่างเสรี ซึ่งความร่วมมือของคนทั่วทุกมุมโลก สามารถเกิดขึ้นได้อย่างสมบูรณ์ เพราะชุมชนในเว็บไซต์ Arduino (https://www.arduino.cc)

และเว็บไซต์นี้ เป็นเครื่องมือพัฒนาฝีมือและสร้างประสบการณ์อันล้ำค่าที่สำคัญของเรา "มือสมัครเล่น" หรือ "ผู้ที่กำลังเริ่มต้นผูกพันกับ Arduino" การ



ที่เราจะเก่งรู้จริงได้ เราจะต้องเข้าใจและเรียนรู้กับปัญหามากมายที่เกิดขึ้นกับการใช้งานบอร์ด Arduino การศึกษาและเรียนรู้ปัญหา คือ ก้าวสำคัญที่เต็มไปด้วยประการณ์อันล้ำค่า "ปัญหาเมื่อไม่ได้เกิดขึ้นกับตัวเองไม่มีทางรู้" และเราจะได้รู้ปัญหาร้อยแปดพันประการ จากประสบการณ์ของคนอื่น และที่สำคัญที่สุด คือ เรียนรู้วิธีแก้ปัญหานั้น

ถ้าเราเข้าไปเยี่ยมชมเว็บไซต์ของ Arduino ก่อนอื่นให้ตรงไปที่ Community หรือหน้าชุมชนของคนพันธุ์ Arduino ในหน้า Forum เราจะพบกับความรู้แบบไร้ขีดจำกัดที่อยู่ในหน้าชุมชนนี้มากมาย ไม่ว่าจะเป็น Installation & Trouble-shooting (การติดตั้งและการแก้ไขปัญหา) Introductory Tutorials (บทเรียนแนะนำ) Project Guidance (แนะนำโครงการโปรเจ็กต์ Arduino) Programming Questions (คำถามการเขียนโปรแกรม) และเรื่องราวที่น่าสนใจอื่นๆ อีกมากมาย แน่นอนปัญหาบางอย่างเราอาจไม่เคยรู้ "เพราะเราไม่เคยเจอ" เช่นนั้นเรียนรู้จากประสบการณ์ของผู้อื่น เพื่อเข้าถึงแนวทางการแก้ไขปัญหาที่ถูกต้อง หากปัญหานั้นเกิดขึ้นกับเราบ้าง

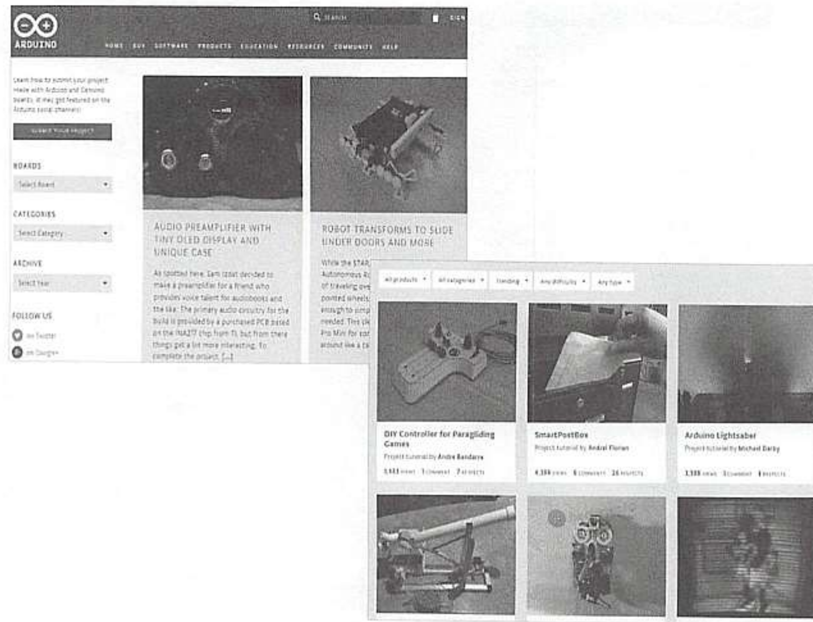
หรือถ้าเราอยากค้นหาไอเดียที่แปลกใหม่ หรืออยากไปเปิดหูเปิดตา เพื่อสร้างแรงบันดาลใจให้กับตัวเอง ให้ไปลองเปิดโลกทัศน์ที่ Blog และ Project Hub ซึ่งทั้งสองหน้านี้ในห้อง Community จะแป็นแหล่งรวบรวมเรื่องราวสุดขั้วและแปลกใหม่สำหรับศาสตร์ของบอร์ด Arduino และที่สำคัญที่สุด สำหรับ

Using Arduino		
Installation & Troubleshooting For problems with Arduino itself, NOT your project Last post: Today at 02:02 pm java.lang.UnsatisfiedLin... by karlsinn	93,903 Posts	21,835 Topics
Introductory Tutorials Tutorials for new people on the forum. Last post: Jun 27, 2018, 12:55 pm Re: Learn how to delete ... by arduiko	336 Posts	46 Topics
Avrdude, stk500, Bootloader issues Problems related to uploading your compiled sketches Last post: Today at 08:23 am error; avrdude: stk500_r... by angelman	1,502 Posts	289 Topics
Project Guidance Advice on general approaches or feasibility Last post: Today at 04:31 pm Re: External 555 watchdo... by wvmarle	479,211 Posts	64,689 Topics
Programming Questions Understanding the language, error messages, etc. Last post: Today at 04:32 pm Re: RS232 projector remo... by PaulS	661,218 Posts	81,039 Topics
General Electronics Resistors, capacitors, breadboards, soldering, etc. Last post: Today at 03:34 pm Re: Question regarding I... by wvmarle	218,882 Posts	21,541 Topics
Microcontrollers Standalone or alternative microcontrollers, in-system programming, bootloaders, etc. Last post: Today at 08:59 am Attiny85 timer control by Frapeno	83,665 Posts	10,155 Topics

หน้า Project Hub เพราะมีโปรเจ็กต์นับพันโครงการที่เป็นประโยชน์สูงสุดของเราชาวมือสมัครเล่น เพราะนัก Arduino ทั่วทุกมุมโลก จะมาโชว์ผลงานโปรเจ็กต์ Arduino ของพวกเขาให้เราได้เรียนรู้ และเอาเป็นแบบอย่าง และแน่นอนพวกเขาไม่หวง Sketch Code หรือถ้าใครมีไอเดียโปรเจ็กต์ Arduino เจ๋งๆ ก็สามารถ Add Project เพื่อโชว์ผลงานของตัวเองได้เช่นเดียวกัน

 Getting Started Project Hub

แน่นอนที่สุด หลักใหญ่ใจความของเว็บไซต์นี้ ไม่ใช่แค่เป็นแหล่งชุมชนของคนพันธุ์ Arduino แต่เว็บไซต์นี้ เปรียบได้กับโปรแกรม Arduino IDE ออนไลน์ ซึ่งเราสามารถสร้างและทดลองโปรเจ็กต์ Arduino ได้ทุกที่ทั่วโลก ที่อาจไม่สะดวกที่จะติดตั้งโปรแกรม Arduino IDE ขอแค่มีอินเตอร์เน็ตก็พอ ไม่ว่าจะเป็นเครื่องโน้ตบุ๊คที่สวนสาธารณะ หรือเครื่องคอมพิวเตอร์ PC ในร้านเกมส์ และนายทุนเงินหนาที่กำลังตัดสินใจจะซื้อลิขสิทธิ์ชุดเซ็นเซอร์อัจฉริยะ "แสดงผลงานให้เขาเห็น" ถ้าจะต้องมัววุ่นกับการติดตั้งโปรแกรม Arduino IDE บนคอมพิวเตอร์ของคนอื่น ก็อาจจะเป็นเรื่องที่ไม่เข้าท่าสักเท่าไหร่ แต่ถ้าเราเข้าใช้



งานเว็บไซต์ Arduino >>> https://www.arduino.cc เราจะสามารถ Run โปรเจ็กต์ Arduino ได้ในทุกที่ทุกเวลา แค่เชื่อมต่อบอร์ด Arduino กับเครื่องคอมพิวเตอร์เท่านั้น

 การลงทะเบียนสร้างบัญชี Arduino

ก่อนเราจะต้องทำการลงทะเบียน เพื่อสร้างบัญชีของเว็บไซต์ Arduino ซึ่งเราจะต้องมีบัญชีอีเมล "ของผู้ให้บริการค่ายไหนก็ได้" ไม่ว่าจะเป็น Outlook หรือ Gmail โดยถ้าเราอยู่ในหน้า Home หรือหน้าไหนๆ ของเว็บไซต์ https://www.arduino.cc ให้สังเกตที่มุมขวาด้านบน จะเห็นช่องทางลงทะเบียน "Sign In" ให้ทำการคลิกเข้าไปเลย

LOGIN
USERNAME
PASSWORD
Forgot your username/password?
LOGIN

CREATE A NEW ACCOUNT
SIGN UP

จากนั้นระบบเว็บไซต์จะให้เราสร้างบัญชีใหม่ (สำหรับคนที่เคยลงทะเบียนไว้แล้ว ก็ Login เข้าใช้งานที่ด้านซ้ายได้เลย) โดยไปที่ช่องด้านขวา Create a New Account และคลิกที่ Sign Up เพื่อเข้าสู่การลงทะเบียนเพื่อสมัครเป็นสมาชิกเว็บไซต์ Arduino

LOGIN

USERNAME

PASSWORD

Forgot your username/password?

LOGIN

CREATE A NEW ACCOUNT

SIGN UP

Sign in with Google

หากลงชื่อเข้าใช้บัญชีใน Browser แล้ว คลิกที่นี่ได้เลย ระบบจะทำการสร้างบัญชีทันที

ระบบจะให้เราใส่ User Name สร้างรหัสผ่าน ยืนยันรหัสผ่าน (ใส่รหัสผ่านอีกครั้งให้เหมือนกัน) ใส่ชื่อบัญชีอีเมล และสุดท้ายยืนยันว่า "ฉันไม่ใช่โปรแกรมอัตโนมัติ"

SIGN UP

ห้ามวรรค ห้ามใส่สัญลักษณ์ใดๆ และจะต้องเป็นภาษาอังกฤษเท่านั้น

CoreFunction

ใส่รหัสผ่าน 2 ครั้ง เพื่อยืนยันรหัสผ่าน

ใส่บัญชีอีเมลของเรา

core.function2017@gmail.com

ฉันไม่ใช่โปรแกรมอัตโนมัติ

reCAPTCHA

ข้อมูลส่วนบุคคล - ข้อกำหนด

คลิกเพื่อยืนยันว่าเราเป็นคนจริงๆ นะ

Subscribe to the Arduino Newsletter

CREATE ACCOUNT



เพื่อให้แน่ใจว่า เราเป็นคนจริงๆ รึเปล่า ระบบจะให้เราตอบคำถามจากภาพ ก็มันให้คลิกรูปอะไร ก็คลิกรูปนั้น คลิกครบแล้ว ก็คลิกที่ "ยืนยัน"

SIGN UP

CoreFunction

core.function2017@gmail.com

ฉันไม่ใช่โปรแกรมอัตโนมัติ

reCAPTCHA

Subscribe to the Arduino Newsletter

CREATE ACCOUNT

ถ้าต้องการได้ข่าวสารหรือสื่อโฆษณาจากเว็บไซต์ Arduino ก็คลิกตรงนี้ได้เลย ระบบเว็บไซต์จะส่งอีเมลข่าวสารต่างๆ ที่เกี่ยวกับวงการ Arduino ไปยังบัญชีอีเมลของเราทั้งวันทั้งคืน

เสร็จแล้วคลิกที่ Create Account

จากนั้นระบบจะส่งอีเมลเพื่อยืนยันตัวบุคคลไปที่บัญชีอีเมลของเรา ให้เราเปิดบัญชีอีเมลของเราขึ้นมา แล้วทำการคลิก Link ในอีเมลของ Arduino Account เมื่อคลิกแล้วแสดงว่า เราได้ทำการลงทะเบียนเป็นสมาชิกของเว็บไซต์ Arduino อย่างสมบูรณ์แล้ว

☐ ☆ Arduino Accounts — Arduino Accounts Activation - Welcome on board! Your username is CoreFunction Please make sure to a — Jun 30

WELCOME
ON BOARD FOLKS!
ARDUINO

คลิกเพื่อยืนยันตัวตน

Your username is CoreFunction

Please make sure to activate your account by clicking the link below:

https://auth.arduino.cc/register/activate/9e38112d-6991-466a-9db4-fcae6306f044

With your brand new account, you can start using our amazing tools to make the most out of your Arduino experience:

- Stay social and follow along with us on Facebook, Twitter, Google+ and Instagram
- Browse other Arduino community members' projects and share yours on the Project Hub
- Program your board using Arduino Web Editor or download the desktop version of the Arduino IDE.
- Have questions? Ask our community's expert users on the Arduino Forum
- Keep up to date with our latest products, events and other happenings! Subscribe to the Arduino official newsletter! Go to arduino.cc and check the footer.

Getting Started Tools

SET UP SOFTWARE



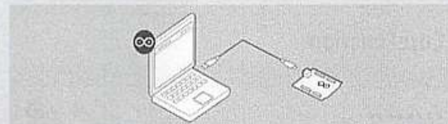
Set up the **Arduino Plugin** to start coding online



Set up **Arduino Cloud**

เรากำลังเข้าสู่ขั้นตอนการเชื่อมต่อบอร์ด Arduino ออนไลน์

Welcome to the Arduino Web Editor Plugin!



The Arduino Web Editor plugin will:

- Upload sketches from the browser onto your boards via a USB cable or the Network
- Allow you to use other Arduino Cloud services

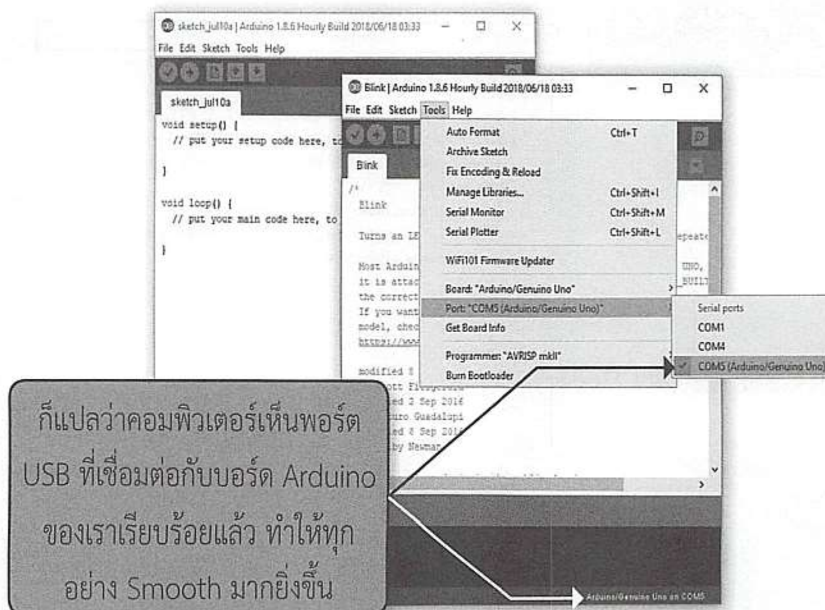
Follow a few simple steps to set up this plugin.

I ALREADY HAVE IT | NEXT

เตรียมพร้อมเสียบสาย USB เพื่อเชื่อมต่อบอร์ด Arduino กับเครื่องคอมพิวเตอร์

 เริ่มต้นการเชื่อมต่อ Arduino ออนไลน์

สำหรับใครที่เคยเชื่อมต่อบอร์ด Arduino กับโปรแกรม Arduino IDE ได้สำเร็จแล้ว จะทำให้ขั้นตอนการเชื่อมต่อบอร์ด Arduino กับเว็บเพจ Arduino Web Editor ง่ายยิ่งขึ้น "ประมาณว่าแทบไม่ต้องทำอะไรเลย ระบบมันคุยกันเองแล้วก็ค้นหากันเจอเอง" เหมือนเป็นบุพเพสันนิวาส ^^!



LOGIN

CoreFunction

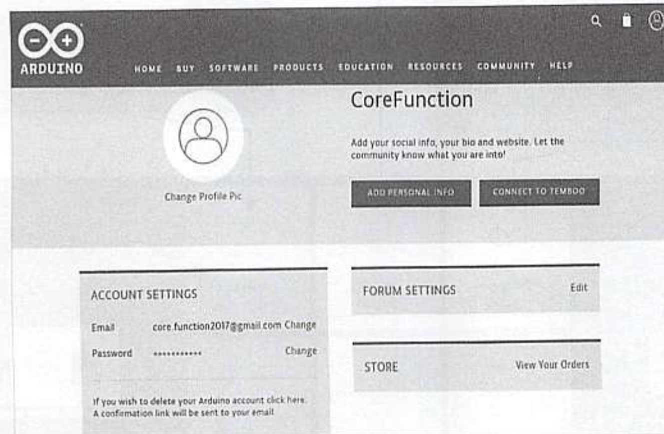
••••••••

Forgot your username/password?

LOGIN

สำหรับการใช้งาน Arduino Web Editor ก่อนอื่นเราจะต้องลงชื่อเข้าใช้ โดยให้คลิกที่ Sign In ที่มุมบนขวามือ จากนั้นใส่ชื่อและรหัสผ่านแล้วคลิกที่ Login

ระบบจะนำเราเข้าสู่หน้า Profile ซึ่งเราสามารถเปลี่ยน หรือแก้ไขข้อมูลส่วนตัวของเราได้ทั้งหมด หรือการเพิ่มเติมและเปลี่ยนแปลงข้อมูลส่วนตัว หรือเพิ่มรูปภาพ Porfile ของเรา

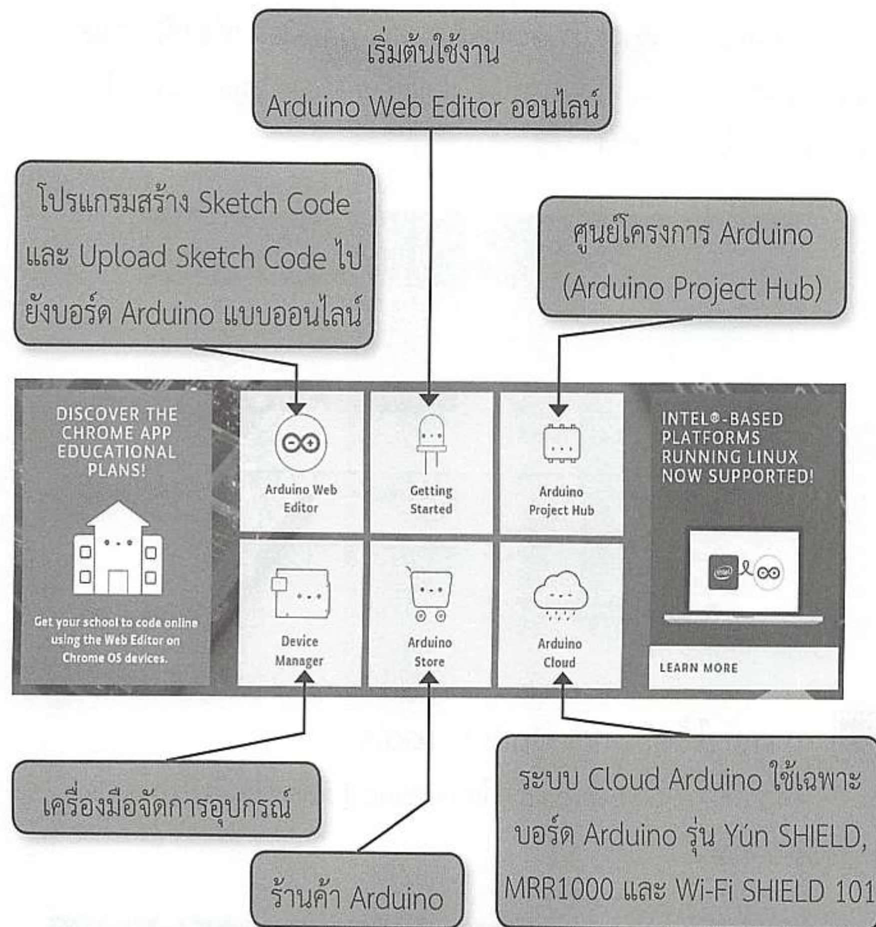


 การเข้าใช้งานหน้า Online Tools

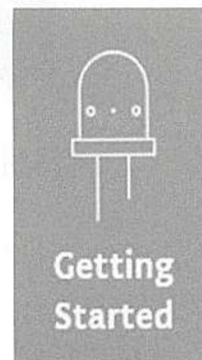
ให้เราเลือกที่ Software แล้วเลือก Online Tools เพื่อเข้าสู่โปรแกรมเครื่องมือออนไลน์

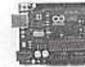


จากนั้นระบบจะนำเราเข้าสู่ ระบบออนไลน์ทั้งหมดของเว็บไซต์ Arduino

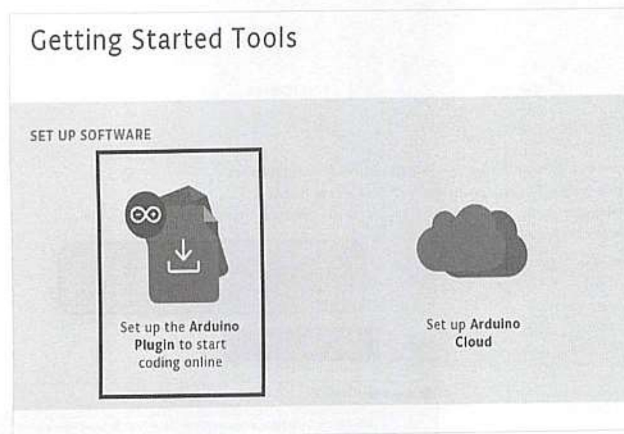


สิ่งที่น่าสนใจที่สุดของเราในตอนนี้ก็คือ Getting Started เพราะเป็นโปรแกรมออนไลน์ ที่สามารถเชื่อมต่อบอร์ด Arduino ได้ หรือเปรียบเสมือนโปรแกรม Arduino IDE ที่ใช้งานได้แบบออนไลน์ นั่นหมายถึง เราสามารถเชื่อมต่อบอร์ด Arduino กับเครื่องคอมพิวเตอร์เครื่องใดก็ได้ หรือเชื่อมต่อที่ไหนก็ได้ ถ้าเช่นนั้นก็คลิกที่ Getting Started ได้เลย

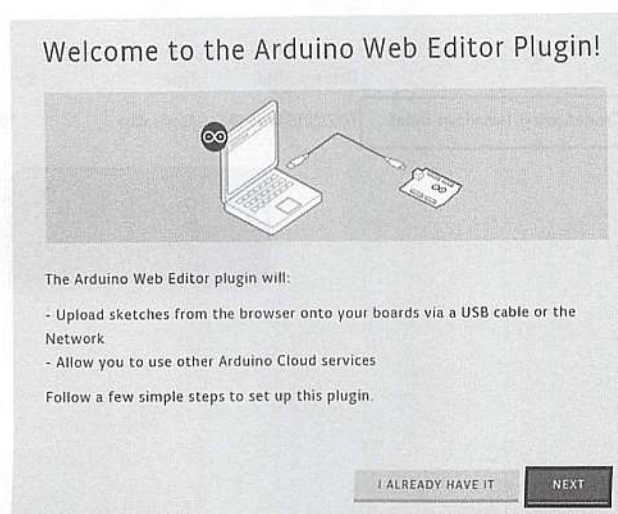


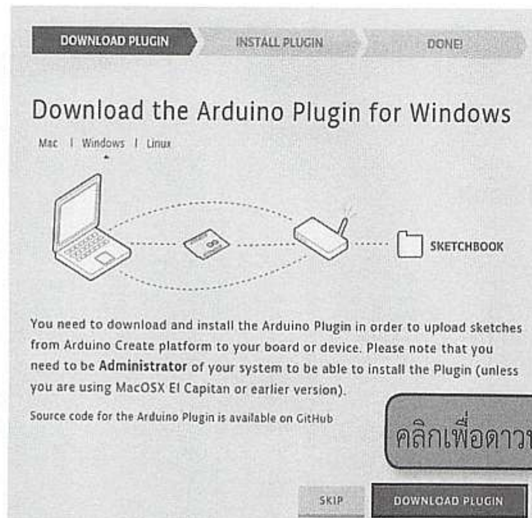


เริ่มแรกเราจะต้องทำการดาวน์โหลดปลั๊กอิน โดยให้คลิกที่ไอคอนด้านซ้าย "Set up the Arduino Plugin"

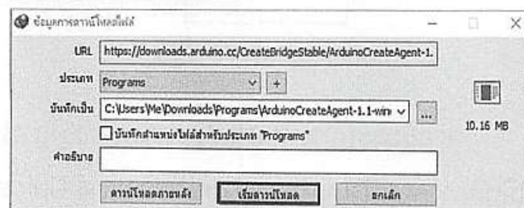


จากนั้นระบบจะอธิบายถึงประโยชน์ของการติดตั้งปลั๊กอินลงบนระบบปฏิบัติการคอมพิวเตอร์ของเรา หมายถึง การ Upload Sketch Code จาก Browser ลงบนบอร์ด Arduino ของเรา ผ่านทางสาย USB (หรือเครือข่าย อันนี้ใช้ในระบบ Cloud ซึ่งใช้ได้สำหรับ Arduino เฉพาะรุ่นที่รองรับเท่านั้น) เช่นนั้นก็คลิกที่ต่อไปได้เลย Next เพื่อทำการดาวน์โหลดปลั๊กอิน





คลิกเพื่อดาวน์โหลดปลั๊กอิน

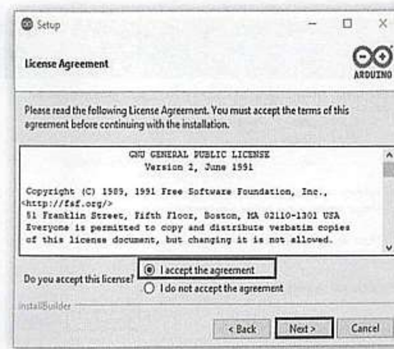


เมื่อระบบดาวน์โหลดปลั๊กอินมาเก็บไว้ในหน่วยความจำคอมพิวเตอร์แล้ว ให้ทำการดับเบิลคลิก เพื่อติดตั้งปลั๊กอิน สำหรับขั้นตอนการติดตั้งก็ไม่ยุ่งยากอะไร

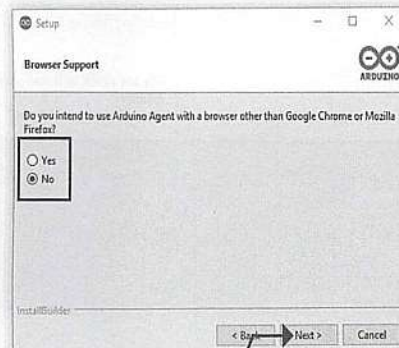
Name	Date modified	Type	Size
ArduinoCreateAgent-1.1-windows-install...	7/12/2018 11:35 PM	Application	10,413 KB

ดับเบิลคลิกเพื่อทำการ Installation





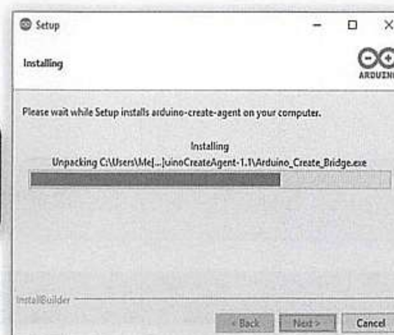




ระบบจะถามเราว่า ต้องการจะใช้ Arduino กับ Browser อื่น นอกเหนือจาก Chrome และ Firefox หรือไม่ ก็ตามความจำเป็นของเราเลย จะ No หรือ Yes จากนั้นคลิกที่ Next

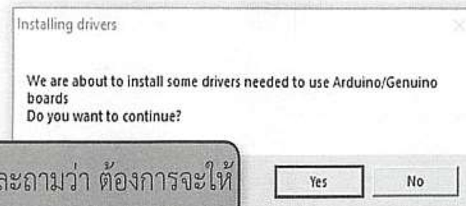


ระบบกำลังดำเนินการติดตั้งปลั๊กอิน



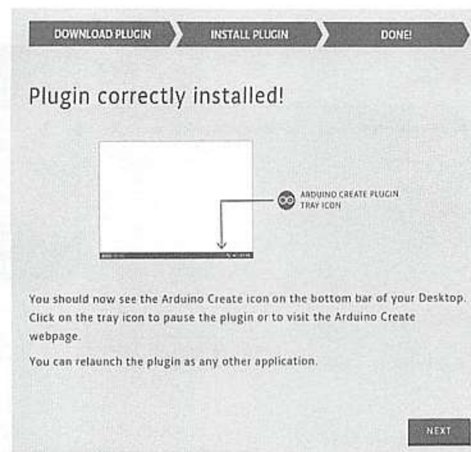






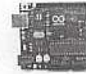
ระบบจะแจ้งถึงสถานะและถามว่า ต้องการจะให้ติดตั้งไดรเวอร์บางอย่างที่จำเป็นในการใช้บอร์ด Arduino หรือไม่ แน่นอนเราก็ Yes ไปเลย

ทีนี้กลับไปที่หน้าเว็บไซต์ที่เปิดหน้าดาวน์โหลดปลั๊กอินทิ้งไว้เมื่อสักครู่ ระบบจะแจ้งว่า จะปรากฏเห็นไอคอนปลั๊กอิน Arduino ที่ Taskbar (ด้านล่างของหน้าจอ)

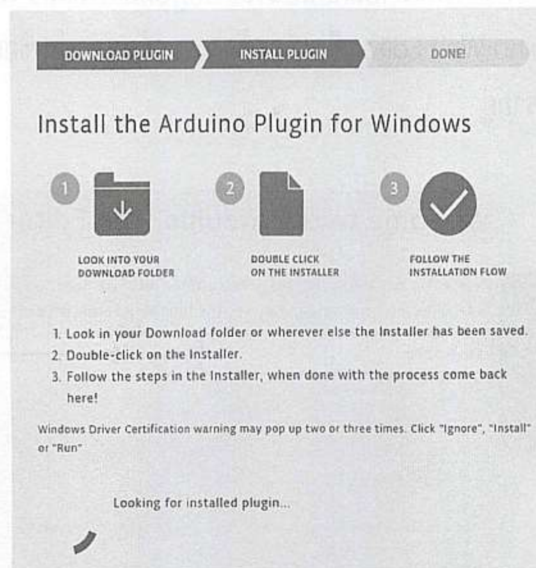


ไอคอนปลั๊กอิน Arduino ที่ Taskbar

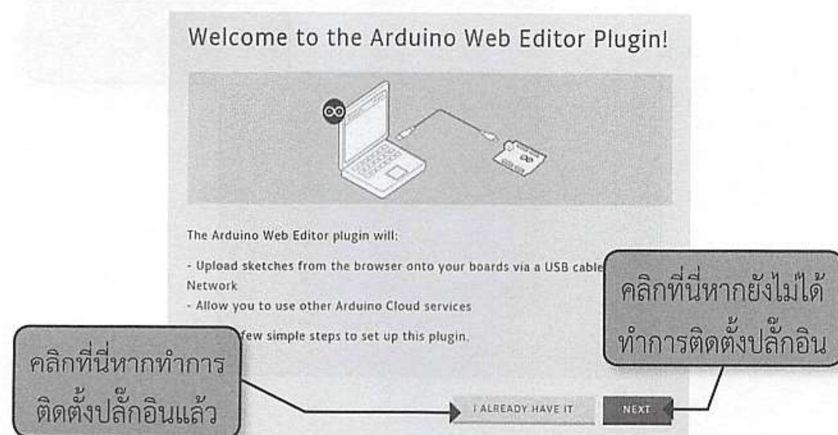


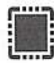


ระบบจะทำการค้นหาปลั๊กอินในระบบคอมพิวเตอร์ของเรา อาจใช้เวลาสักครู่



หรือไม่ว่าจะเป็นการย้อนกลับหน้าเพจด้วยเหตุผลประการใดก็ตาม หากเราทำการติดตั้งปลั๊กอินบนระบบปฏิบัติการคอมพิวเตอร์แล้ว และให้คลิกที่ "I Already Have It" หมายถึง เราติดตั้งปลั๊กอินบนระบบปฏิบัติการแล้ว



 การใช้งาน Arduino Web Editor "Arduino IDE ออนไลน์"

เมื่อระบบทำการค้นหาปลั๊กอินบนระบบปฏิบัติการได้แล้ว ระบบจะพาเราเข้าสู่หน้า Arduino Web Editor ซึ่งเราจะต้องกรอกข้อมูล เช่น ชื่อและนามสกุล และประเทศที่เราอยู่

Welcome to the Arduino Web Editor

Before you can start using the Arduino Web Editor we need you to provide us with some informations, and accept the following Terms & Condition:

ชื่อหรือนามสกุล อะไรก็ได้ ไม่จำเป็นต้องใช้ของจริง

Core Function

Thailand

Thailand

BCMI Labs SA. Create Terms of Service

Last updated: May 20, 2016

Please read these Terms of Service carefully before using the create.arduino.cc website.

1. Acceptance of Terms

By using the Arduino Create Web site (the "Site") in any way, including using, transmitting, downloading, or uploading any of the services or functionality (the "Service") made available or enabled via the Site by BCMI Labs SA, or merely browsing the Site, you agree to these Terms of Service. You may not use the Service, or accept these Terms of Service, if (a) you are not of legal age to form a binding contract with BCMI Labs SA; or (b) you are prohibited by law from receiving or using the Service. If you are entering into these Terms of Service on behalf of a company or other legal entity, you represent that you have the authority to bind such entity to these Terms of Service, in which case "you" or "your" shall refer to such entity. BCMI Labs SA makes the Service available only if you have registered for an account. The Service may also be subject to a subscription or other agreement, posted guidelines, rules or terms of

สมัครรับจดหมายข่าว

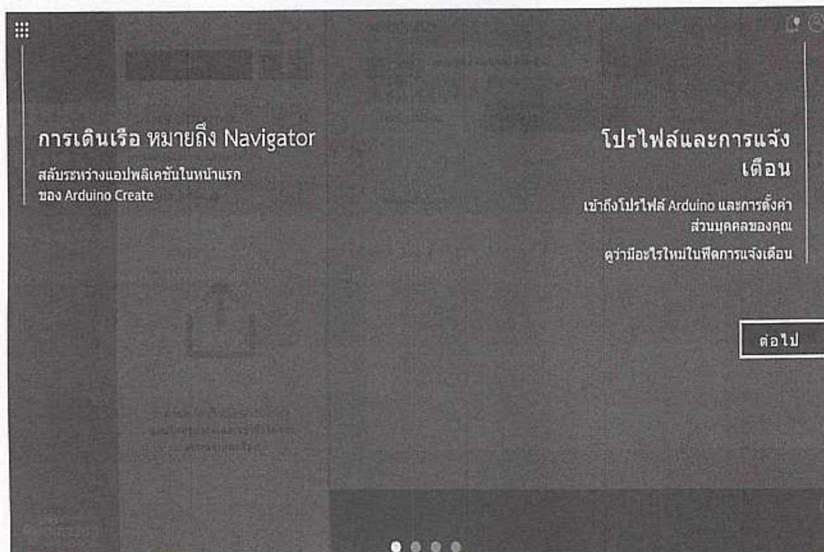
Sign up to the Arduino Create Newsletter

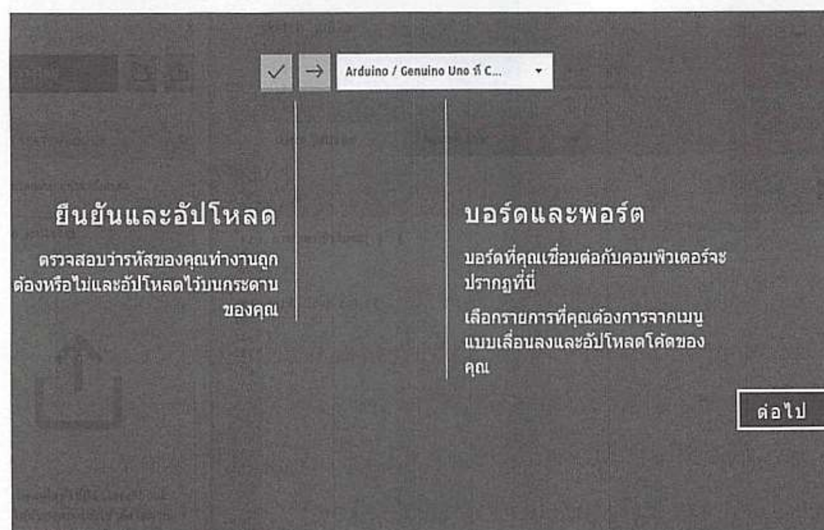
DECLINE AGREE

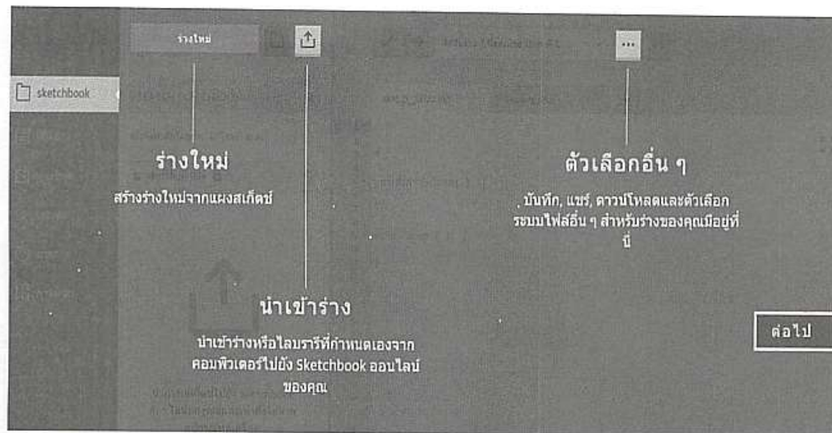
เสร็จแล้วคลิกที่ Agree

Loading Arduino Create...

จากนั้นระบบจะเริ่มต้นด้วยการพาทัวร์เมนูการใช้งานต่างๆ ของ Arduino Web Editor ถ้าใช้ Browser ที่มี Extensions แปลภาษา ก็ไม่ต้องอธิบายกันให้เสียเวลา ไปดูพร้อมกันเลย

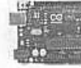








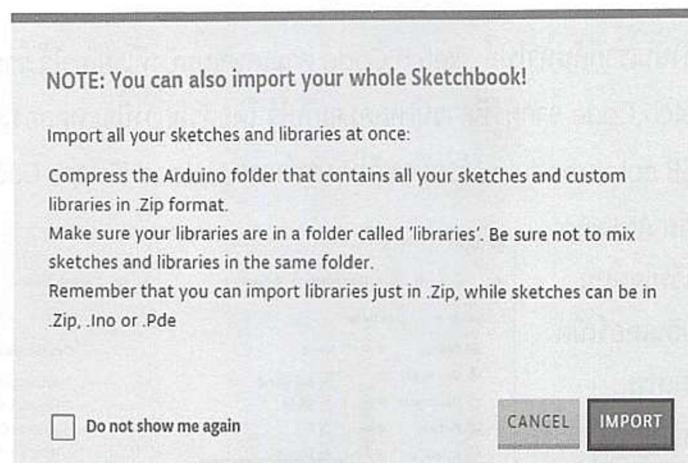
จากนั้นระบบจะถามเราว่า ต้องการนำไฟล์ Sketch Code ที่เราสร้างเอาไว้แล้ว เข้าสู่ระบบ Arduino Web Editor หรือไม่ ถ้าเรามี Sketch Code อยู่แล้วก็คลิก Yes! ไปได้เลย (หมายถึง ถ้าได้ทำการสร้างไฟล์ Sketch Code เก็บไว้ในพื้นที่จัดเก็บของโปรแกรม Arduino IDE แล้ว ไฟล์นามสกุล ino เช่น Heartbeat_Sensor_Module.ino หรือไฟล์ Sketch Code ที่เคยดาวน์โหลดเก็บไว้ ก็สามารถคลิก Yes เพื่อ Import ไฟล์นั้นได้เลย)



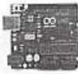


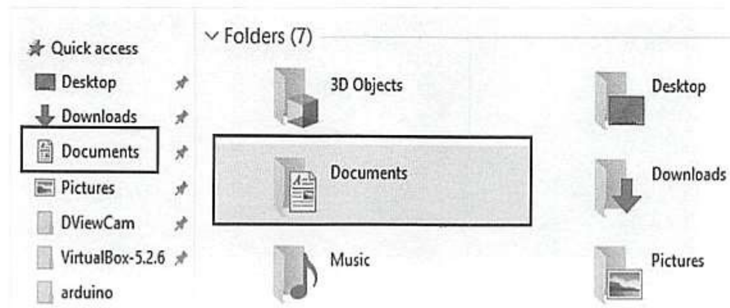
การนำเข้า Sketch Code ที่สร้างไว้ในโปรแกรม Arduino IDE หรือดาวน์โหลดไฟล์ Sketch Code ไว้ก่อนแล้ว

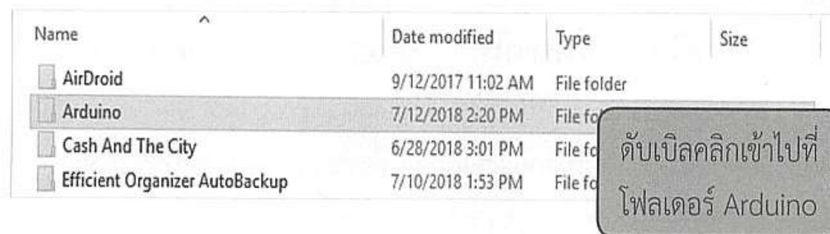
คลัง Sketch ของระบบเว็บไซต์ Arduino Web Editor จะเรียกว่า Sketchbook ให้เราคลิกที่ Import ได้เลย เพื่อเริ่มต้นการนำเข้าไฟล์ Sketch Code ที่เราได้สร้างไว้แล้วกับโปรแกรม Arduino IDE



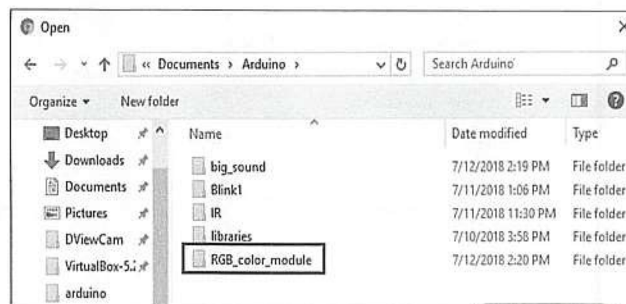
จากนั้นค้นหาไฟล์ Sketch ที่ถูกบันทึก (Save) โดยโปรแกรม Arduino IDE อันนี้ถ้าใครไม่สังเกตก็จะตกม้าตายได้ง่ายๆ จำเป็นจะต้องเขียน Sketch Code ใหม่ซ้ำสองบนเว็บ Arduino Web Editor (เสียเวลา) เช่นนั้นจะบอกที่อยู่ซึ่งเป็นคลังไฟล์ Sketch Code ของโปรแกรม Arduino IDE ในหน่วยความจำคอมพิวเตอร์ โดยให้ไปที่ Document >>> Arduino

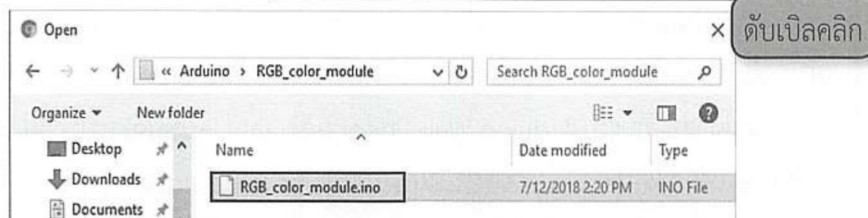




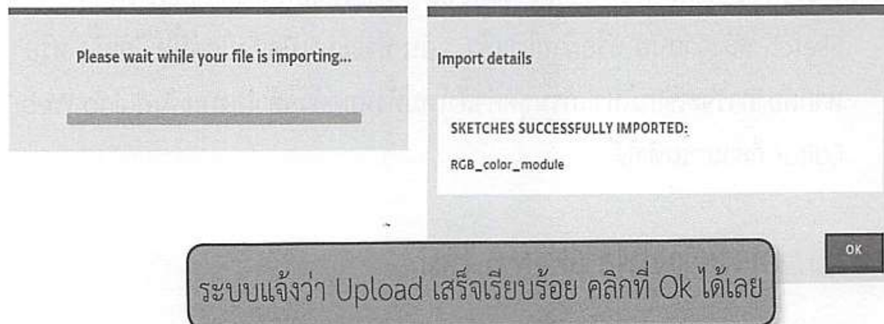


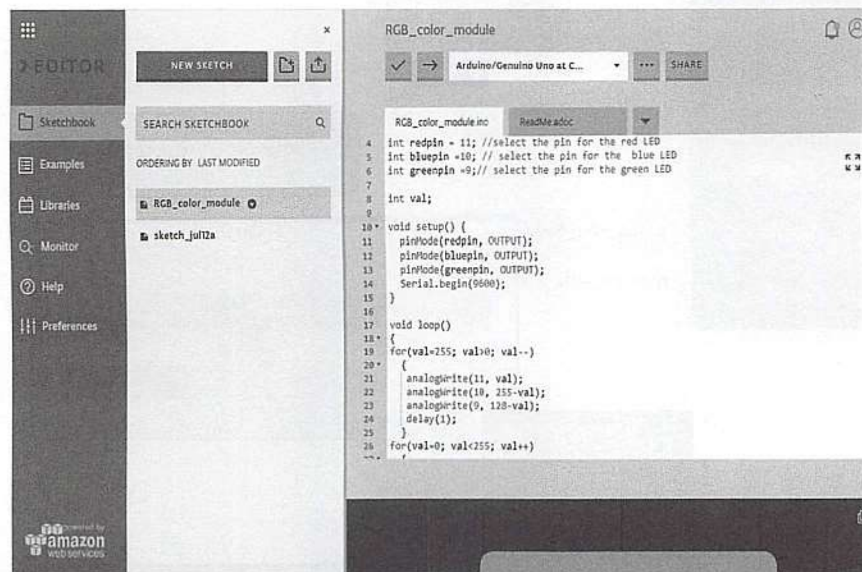
ดังนั้นเราก็ค้นหาไฟล์ Sketch Code ตามเศษขนมปังที่โรยเอาไว้ รวมถึงไฟล์ Sketch Code จากผู้รู้ที่ดาวน์โหลดมาเก็บไว้ โดยตัวอย่างนี้จะทดลองใช้ชุดคำสั่ง RGB color module (ตัวอย่างเดิม) มาทำการ Upload Sketch Code ไปยังบอร์ด Arduino ผ่านทางเว็บไซต์บนเครือข่ายอินเตอร์เน็ต (ไม่ใช่โปรแกรม Arduino IDE นะ)





ระบบจะทำการ Upload ไฟล์ Sketch Code จากหน่วยความจำคอมพิวเตอร์ มาเก็บไว้ยังเว็บไซต์ Arduino Web Editor ทันที อาจใช้เวลาเพียงชั่วครู่



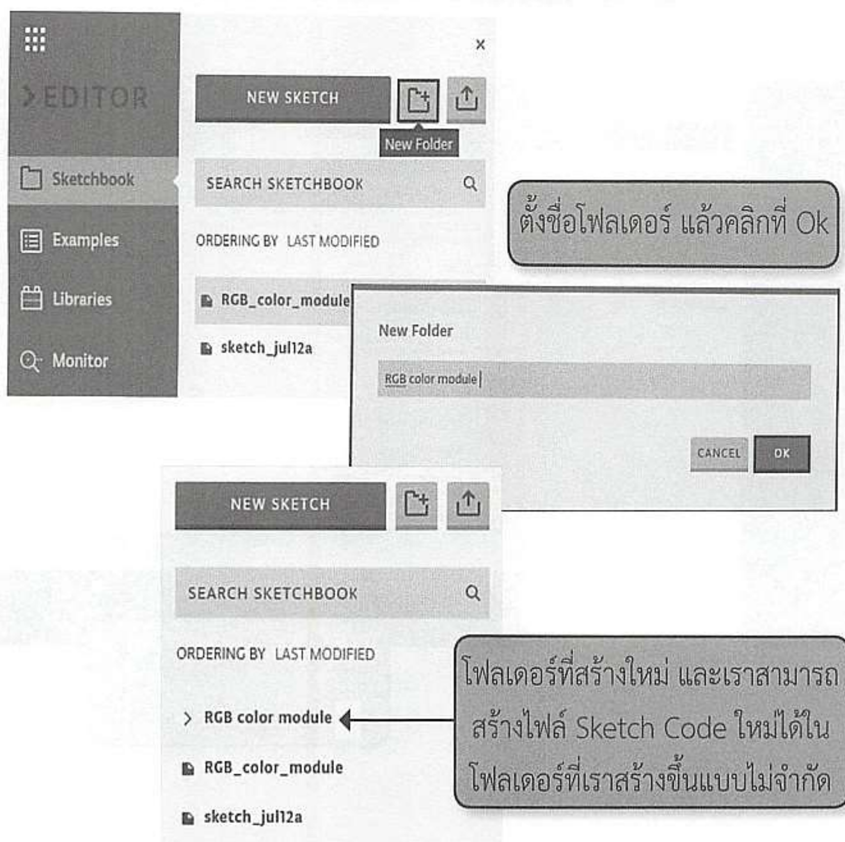


Sketch Code จะแสดงบน Arduino Editor ทันที

 การสร้างไฟล์ Sketch ใหม่ และการนำเข้าไฟล์ Sketch โดยช่องทางปกติ

"เดี๋ยวก่อน!" เมื่อสักครู่เป็น Import หลังจากที่เข้าสู่การใช้งานของเว็บไซต์ Arduino Web Editor ในครั้งแรก หากเราต้องการ Import ไฟล์ Sketch หลังจากนั้น หรือผ่านไปแล้ว 3 วัน พึ่งจะมาเปิดเว็บไซต์นี้ใช้อีกครั้ง หรือหากต้องการจะเขียนโปรแกรมชุดคำสั่งใหม่ทั้งหมดด้วยตัวเองบน Arduino Web Editor ก็สามารถทำได้

การสร้างไฟล์ Sketch ใหม่



EDITOR

Sketchbook

Examples

Libraries

Monitor

NEW SKETCH

New Folder

SEARCH SKETCHBOOK

ORDERING BY LAST MODIFIED

RGB_color_module

sketch_jul12a

เช่น สร้างโฟลเดอร์ ในชื่อ Project Sensor ซึ่งแน่นอนจะมีหลายโปรเจ็กต์ที่เราจะต้องสร้างไฟล Sketch Code ขึ้นใหม่

New Folder

Project Sensor

CANCEL OK

NEW SKETCH

SEARCH SKETCHBOOK

ORDERING BY LAST MODIFIED

> Project Sensor

นี่คือโฟลเดอร์ที่เราสร้างขึ้นใหม่

> SMD RGB Module (1)

> Sound Sensor Module (2)

RGB_color_module

sketch_jul12a

NEW SKETCH

SEARCH SKETCHBOOK

ORDERING BY LAST MODIFIED

> Project Sensor

Duplicate Folder

Rename Folder...

Move Folder to...

Delete Folder...

New Folder

New Sketch

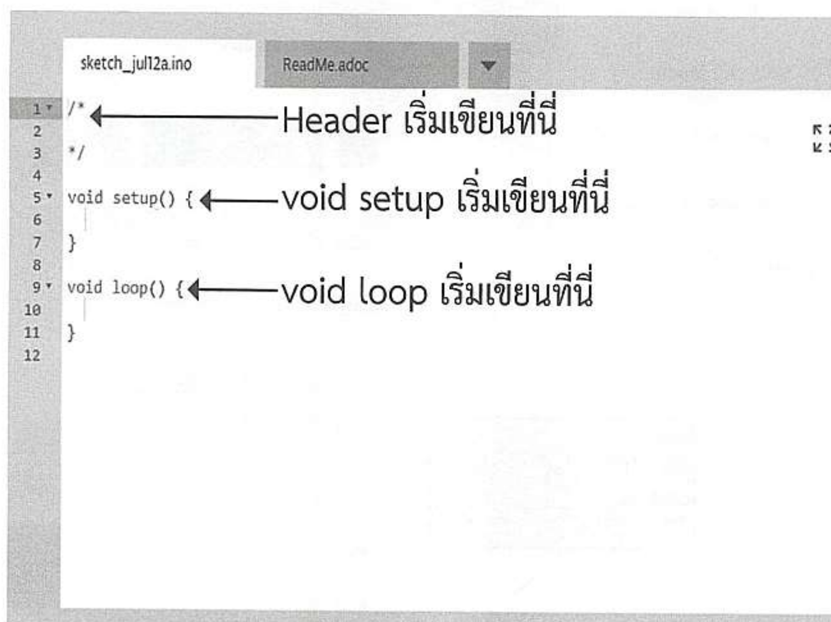
คลิกขวาที่โฟลเดอร์แล้วเลือก New Sketch

ไม่ว่าจะเป็นโปรเจ็กต์ Sensor ไหน เราก็สามารถสร้าง Sketch Code ในโฟลเดอร์นี้ได้แบบไม่จำกัด (จะกี่โปรเจ็กต์ก็ได้) แค่คลิกขวาบนโฟลเดอร์ และเลือก New Sketch

ทีนี่กลับมาที่ตัวอย่างเดิม RGB collor module ให้ทำการคลิกขวาโฟลเดอร์ที่สร้างใหม่ แล้วเลือก New Sketch



ระบบจะสร้างไฟล์ Sketch ขึ้นมาใหม่ พร้อมด้วยชุดคำสั่งเริ่มต้นเช่นเดียวกับการสร้าง Sketch ใหม่ บนโปรแกรม Arduino IDE



เราสามารถเขียนโปรแกรมใหม่ เปลี่ยนแปลง ลบ Copy Paste ได้หมดทุกอย่าง แต่การ Paste หรือวาง จะต้องใช้วิธี Ctrl + V อย่างเดียว คลิกขวาแล้วเลือก Paste ไม่ได้

sketch_jul12a.ino ReadMe.adoc

```
 */
int redpin = 11; //select the pin for the red LED
int bluepin =10; // select the pin for the  blue LED
int greenpin =9;// select the pin for the green LED

int val;

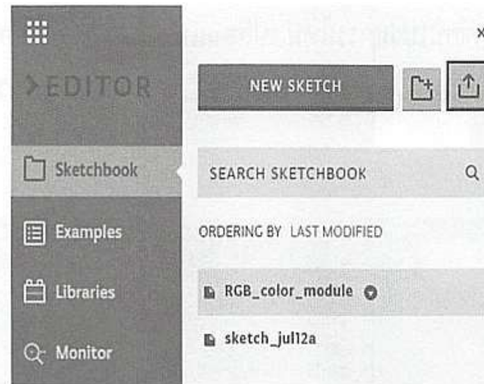
void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
for(val=255; val>0; val--)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
```

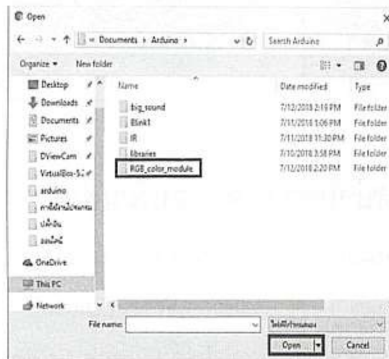
ถ้า Copy และ Paste ทั้งหมด อย่าลืมลบค่าเบื้องต้นของไฟล์ Sketch ก่อน คือ ส่วนของ Header ส่วนของ void setup() ส่วนของ void loop() และเครื่องหมายและสัญลักษณ์อื่นๆ ทั้งหมด (เดี๋ยวระบบจะงงมีคำสั่งซ้ำกัน)

 การนำเข้าไฟล์ Sketch จากหน่วยความจำคอมพิวเตอร์

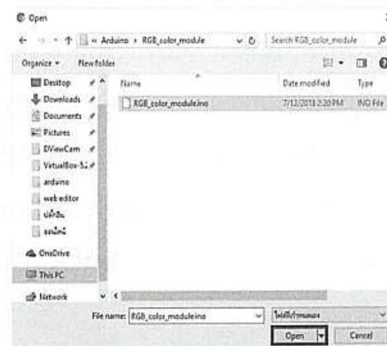
สำหรับวิธีการนำเข้าก็เหมือนกันกับตอนแรกที่เปิดเข้าใช้ Arduino Web Editor เพียงแต่ทางเข้าต่างกัน เพราะการเปิดเข้าใจใช้ Arduino Web Editor ครั้งแรก ระบบจะเปิดช่องทางพิเศษ เพื่อให้เราสามารถนำเข้าไฟล์ Sketch Code ที่มีอยู่ในหน่วยความจำคอมพิวเตอร์อยู่แล้ว มาเก็บไว้บน Web Editor แต่หากเป็นการใช้งานเว็บไซต์ออนไลน์ครั้งต่อๆ ไป เราจะสามารถนำเข้า หรือ Import ไฟล์ Sketch Code ได้ที่ปุ่มลูกศรชี้ขึ้น หรือไอคอน Upload (แต่เป็นคนละปุ่มกับไอคอน Upload โปรแกรมไปยังบอร์ด Arduino นะ)



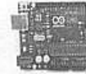
จากนั้นระบบจะเข้าสู่เข้าตอนการเลือกไฟล์ Sketch Code โดยให้เราไปที่โฟลเดอร์เก็บ Sketch ของโปรแกรม Arduino IDE ในหน่วยความจำคอมพิวเตอร์ ซึ่งเป็นขั้นตอนที่เหมือนกันกับการ Import Sketch Code ในตอนแรก



คลิกเลือกโฟลเดอร์ที่เก็บไฟล์ Sketch แล้วคลิกที่ปุ่ม Open

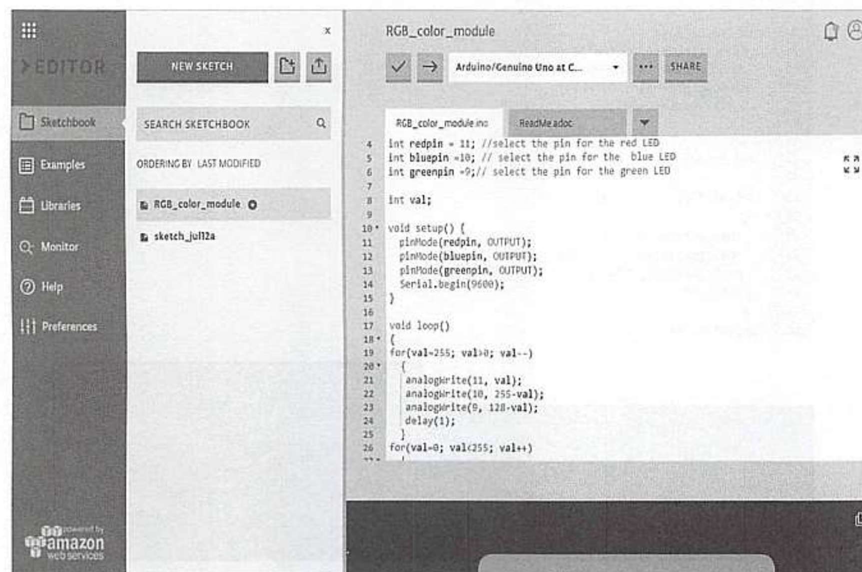


คลิกไฟล์ Sketch (ไฟล์นามสกุล .ino) จากนั้นคลิกที่ปุ่ม Open

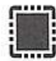


ระบบจะทำการ Upload ไฟล์ Sketch Code จากหน่วยความจำคอมพิวเตอร์ มาเก็บไว้ยังเว็บไซต์ Arduino Web Editor ทันที อาจใช้เวลาเพียงชั่วครู่





Sketch Code จะแสดงบน Arduino Editor ทันที

 การ Verify หรือ Compile และ Upload ไปยังบอร์ด Arduino แบบออนไลน์

เมื่อทำการสร้าง เขียน หรือ Copy โปรแกรมชุดคำสั่งเรียบร้อยแล้ว ต่อไปเราก็ทำการ Verify หรือ Compile เพื่อตรวจสอบ รวบรวม ประมวล และยืนยัน Sketch Code ก่อนที่จะทำการ Upload ไปยังบอร์ด Arduino

RGB_color_module

Arduino/Genuino Uno at C... SHA Verifying RGB_color_module

Verify

RGB_color_module.ino ReadMe.adoc

```
int redpin = 11; //select the pin for the red LED
int bluepin =10; // select the pin for the  blue LED
int greenpin =9;// select the pin for the green LED

int val;

void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
for(val=255; val>0; val--)
  {
   analogWrite(11, val);
   analogWrite(10, 255-val);
   analogWrite(9, 128-val);
   delay(1);
  }
for(val=0; val<255; val++)
```

ระบบทำการ Verify เรียบร้อยแล้ว

Success: Done verifying RGB_color_module

```
Sketch uses 2474 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 190 bytes (9%) of dynamic memory, leaving 1858 bytes for local
variables. Maximum is 2048 bytes.
```

จากนั้นให้ทำการ Upload Sketch Code ไปยังบอร์ด Arduino ที่ทำการเชื่อมต่อวงจรโมดูลอิเล็กทรอนิกส์เรียบร้อยแล้ว ผ่านเว็บไซต์ Arduino Web Editor บนเครือข่ายอินเตอร์เน็ต

RGB_color_module

Arduino/Genuino Uno at C... ··· SHARE

Upload

จากนั้นคลิกที่ Upload

ระบบกำลังทำการ Upload

Uploading RGB_color_module

```
Sketch uses 2474 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 190 bytes (9%) of dynamic memory, leaving 1858 bytes for local
variables. Maximum is 2048 bytes.
```

ระบบ Upload สำเร็จแล้ว

Success: Done uploading RGB_color_module

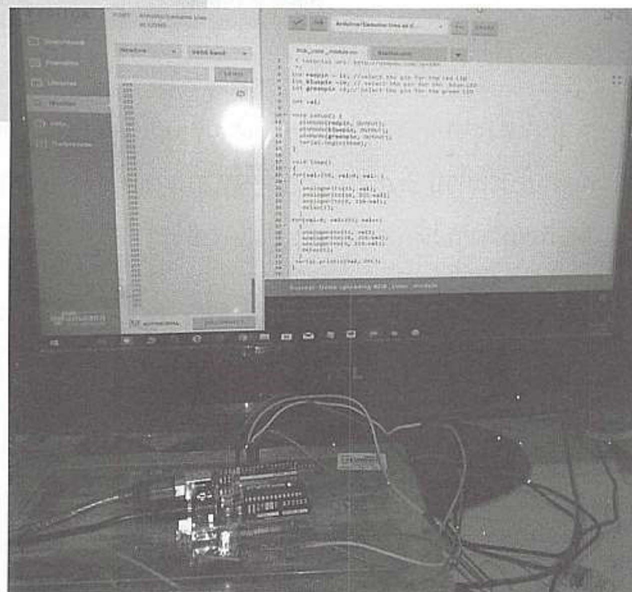
```
avrdude: 2474 bytes of flash verified
avrdude: safemode: Fuses OK (E:00, H:00, L:00)
avrdude done. Thank you.
```

ทันทีที่ระบบแจ้งว่า Success: Done uploading บอร์ด Arduino จะทำงานตามโปรแกรม Sketch Code ซึ่งหากเราต่อวงจรอิเล็กทรอนิกส์ถูกต้อง ชุดโมดูลก็จะทำงานได้ตามที่เราตั้งโปรแกรมไว้

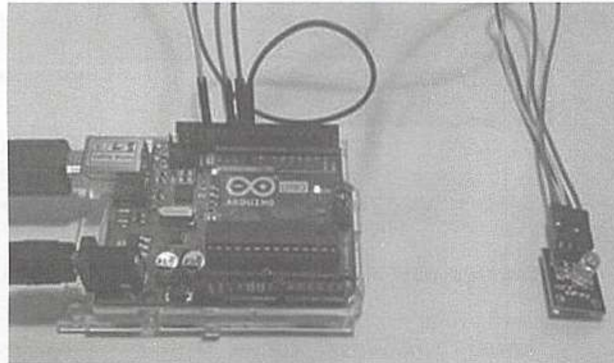
หากเราคลิกที่เมนูด้านซ้ายมือของเรา "Monitor" ระบบจะแสดงค่าการตอบสนองที่บอร์ด Arduino ส่ง-รับสัญญาณไปยังโมดูลอิเล็กทรอนิกส์ ตามตัวอย่าง คือ RGB collor module ซึ่งระบบจะแสดงค่าการติด-ดับของไฟ LED ตามเวลาที่ได้กำหนดในโปรแกรม

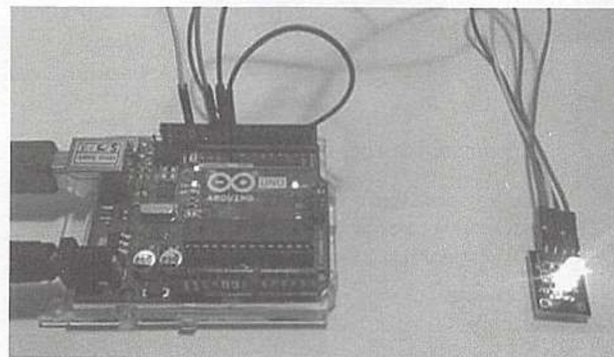


Run Sketch Code ผ่านทางเว็บไซต์ Arduino Web Editor บนเครือข่ายอินเตอร์เน็ต สุดยอดดด!




โดยมอนิเตอร์จะแสดงค่าอะนาล็อก
ของแสงไฟ LED ในหลายระดับที่เรา
ได้กำหนดไว้ คือ 225 (val-225)





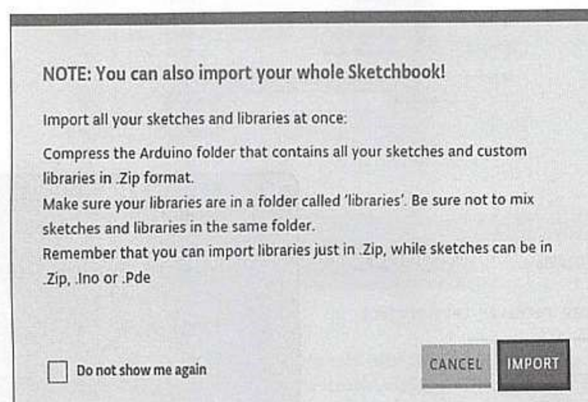
 การนำเข้า Libraries ของ Arduino Web Editor

เช่นเดียวกับโปรแกรม Arduino IDE ถ้าการ Upload Sketch Code ไปยังบอร์ด Arduino ไม่มี Libraries ของ Sketch Code ซึ่งเป็นชุดคำสั่งของโมดูลอิเล็กทรอนิกส์ ที่เราต้องต่อวงจรเชื่อมต่อกับบอร์ด Arduino โปรแกรม Arduino IDE ย่อมไม่สามารถ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino ได้สำเร็จ "Arduino Web Editor" ก็เหมือนกัน เช่นนั้นในเว็บไซต์ จึงมีช่องทางเพื่อให้เราสามารถ Upload Libraries จากหน่วยความจำคอมพิวเตอร์ของ Sketch Code นั้นๆ มาเก็บไว้ในคลัง Libraries ของเว็บไซต์ โดยขั้นตอนแรกเราจะต้องไปที่เมนูด้านซ้ายมือเรา แล้วคลิกที่เมนู Libraries

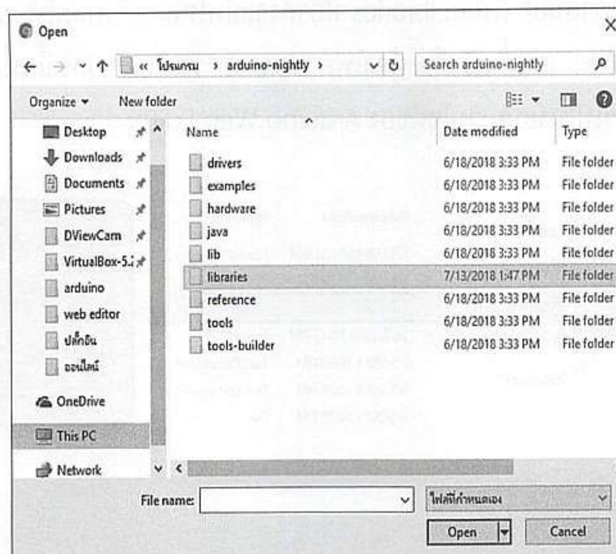
โดยในเว็บไซต์ ก็จะมี Libraries อยู่ในคลังมากพอสมควร อาจจะมากกว่าในโปรแกรม Arduino IDE เสียด้วยซ้ำ "แต่ยังไงล่ะ" บางที Sketch Code ที่เราใช้ในเว็บไซต์ดันไม่มี Libraries

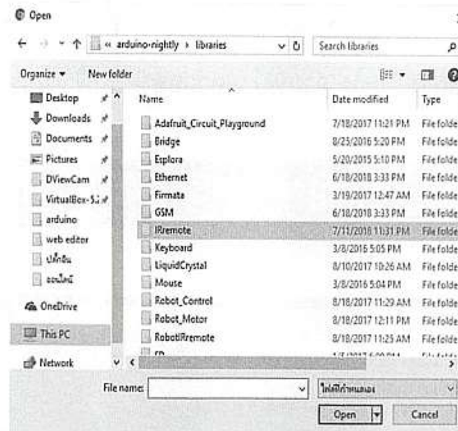
เช่นนั้นให้เราทำการคลิกที่ไอคอน Upload เพื่อ Import หรือนำเข้า Libraries ของ Sketch Code จากหน่วยความจำคอมพิวเตอร์





จากนั้นให้เข้าไปที่โฟลเดอร์ที่เก็บไฟล์โปรแกรม Arduino IDE แล้วเข้าไปที่โฟลเดอร์ Libraries ต่อจากนั้นก็เลือกไฟล์ Libraries ที่เราต้องการ





sketch_jul18e.ino   ReadMe.adoc

```
* IR remote receiver test project
 */
#include <IRremote.h> //adds the library
const int irReceiverPin = 12; //receiver
const int ledPin = 2;
IRrecv irrecv(irReceiverPin);
decode_results decodedSignal;
```

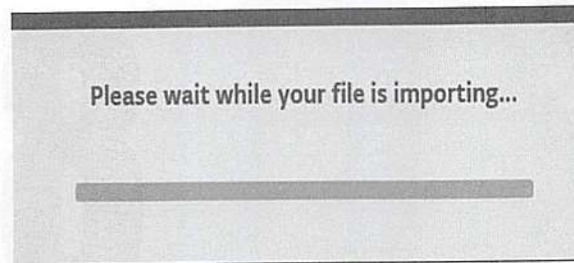
สำคัญที่สุด เลือกไฟล์ที่มีชื่อเดียวกัน นามสกุลเดียวกันตามที่ได้มีการประกาศคำสั่งพิเศษไว้ใน Sketch Code ในส่วนของ Header

เช่น ตามตัวอย่างได้เขียนคำสั่งพิเศษไว้ในส่วนของ Header เพื่อให้ระบบเรียกใช้ไฟล์ภายนอก (ไฟล์ Libraries ที่กำลังจะอัพโหลด) กำหนดให้ #include <IRremote.h> เช่นนั้นเราก็จะต้องไฟล์ Libraries ที่มีชื่อเดียวกันนี้ เพื่อทำการอัพโหลดเข้าไปในระบบเว็บไซต์ของ Arduino Web Editor

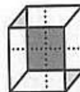
Name	Date modified	Type
examples	7/11/2018 11:31 PM	File folder
IRremote.cpp	5/5/2011 10:28 PM	CPP File
IRremote.h	5/5/2011 10:28 PM	H File
IRremoteInt.h	12/8/2014 10:17 PM	H File
keywords	5/5/2011 10:28 PM	Text Document
LICENSE	5/5/2011 10:28 PM	Text Document
readme	5/5/2011 10:28 PM	File

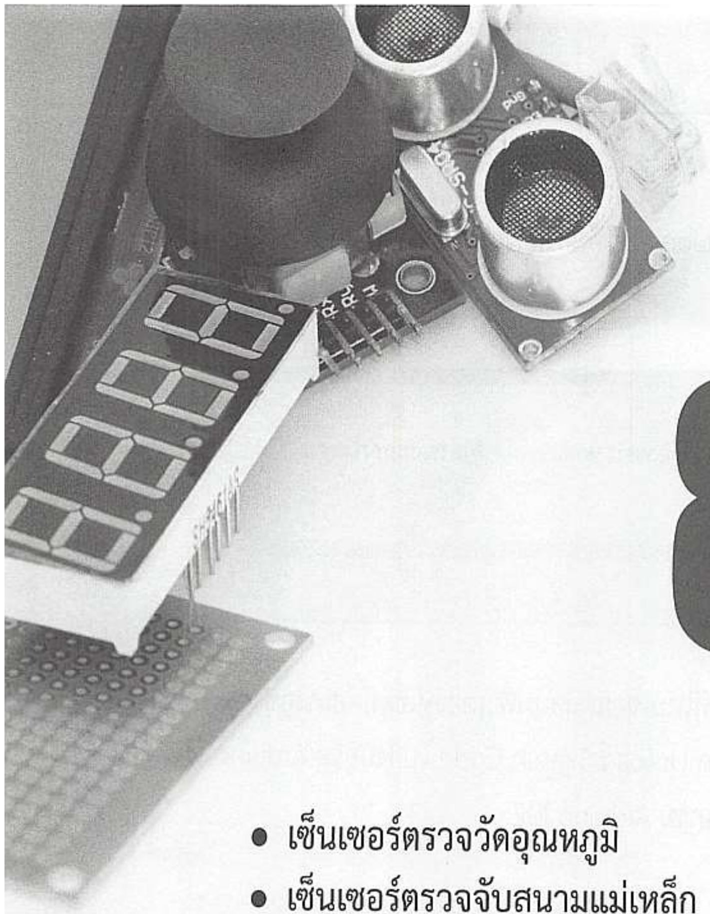
คลิกเลือกที่ไฟล์นี้ ซึ่งมีชื่อเหมือนกับในคำสั่งพิเศษ #include <IRremote.h>

เมื่อเลือกไฟล์ Libraries ที่ต้องการแล้ว ระบบจะทำการ Upload ไปยังเว็บไซต์ทันที



ไม่ว่าเราจะอยู่ที่ไหน ขอแค่มีคอมพิวเตอร์ที่เชื่อมต่อเครือข่ายระบบอินเตอร์เน็ต ก็สามารถ Upload Sketch Code ไปยังบอร์ด Arduino ได้ โดยไม่จำเป็นต้องอาศัยโปรแกรม Arduino IDE





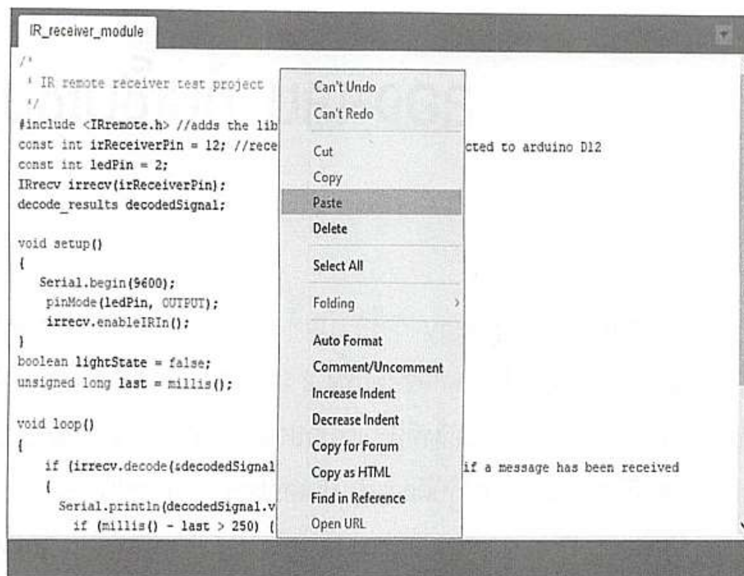
8

- เซ็นเซอร์ตรวจวัดอุณหภูมิ
- เซ็นเซอร์ตรวจจับสนามแม่เหล็ก
- ตรวจจับโลหะ
- เซ็นเซอร์ตรวจจับเสียง
- เซ็นเซอร์ตรวจวัดค่าอุณหภูมิและความชื้นในอากาศ
- ตรวจจับความเคลื่อนไหว
- เซ็นเซอร์ตรวจจับเปลวไฟด้วยแสงอินฟราเรด
- ตรวจจับพื้นหรือเส้นสีขาวและสีดำ
- เซ็นเซอร์ที่ตรวจจับสิ่งกีดขวาง
- โมดูลส่งเสียงสัญญาณเตือน
- ชุดไฟ LED
- ตรวจวัดสัญญาณชีพ
- เซ็นเซอร์ตรวจจับการสั่นสะเทือน
- Joystick เครื่องมือเล่นเกมยอดฮิต

ลอง-เล่น กับเซ็นเซอร์

จุดเด่นของบอร์ด Raspberry Pi คือ เป็นทั้งระบบปฏิบัติการคอมพิวเตอร์และเป็นทั้งระบบไมโครคอนโทรลเลอร์ โดยทั้ง 2 ระบบอยู่ในบอร์ดเดียวกัน และมีขนาดเท่ากับบอร์ด Arduino แต่สิ่งที่เป็นเหมือนข้อเสียเพียงเล็กน้อยของบอร์ด Raspberry Pi ก็คือ เป็นระบบปฏิบัติการซึ่งเป็นเอกเทศ แม้เราจะใช้ Remote Desktop, Desktop Simulator, SSH puTTY หรือ VNC Viewer ซึ่งสามารถใช้ได้พร้อมกันกับระบบปฏิบัติการหลัก เช่น Windows ได้ก็ตาม “ใช้บนจอมอนิเตอร์เดียวกัน ใช้เมาส์ร่วมกัน คีย์บอร์ดเดียวกัน” แต่ถึงอย่างไรก็เป็นคนละระบบปฏิบัติการอยู่ดี นั่นแปลว่า เราไม่สามารถ Copy Code Python จากเว็บไซต์บน Browser ของระบบปฏิบัติการ Windows ไป Paste หรือวางลงในโปรแกรม Python บนระบบปฏิบัติการ Raspbian ของบอร์ด Raspberry Pi ทั้งที่หน้าจอเดียวกัน เมาส์ใช้ตัวเดียวกัน คีย์บอร์ดใช้ร่วมกัน “แต่มันเป็นคนละระบบปฏิบัติการไง” แน่นอนมันเป็นอุปสรรคสำหรับมือใหม่ที่ไม่ค่อยมีแรงบันดาลใจหรือไม่ค่อยจะมีความพยายามในการใช้บอร์ดไมโครคอนโทรลเลอร์ “อยากใช้ได้ใช้เลย อยากใช้ได้ใช้เป็นทันที ขี้เกียจจำ และที่สำคัญขี้เกียจพิมพ์” นอกเสียจากเราจะต้องมีมอนิเตอร์ของระบบปฏิบัติการ Raspberry Pi แยกออกต่างหากจากระบบปฏิบัติการ Windows ดังนั้นจอมอนิเตอร์จะต้องมี HDMI Port หรือ HDMI

to VGA Converter Adaptor เราจึงจะสามารถ Copy Python Code ไป Paste หรือวางในโปรแกรม Python ได้ทันที "อันนี้ไม่มีปัญหา" แต่ Browser Chromium ของระบบปฏิบัติการ Raspbian บนบอร์ด Raspberry Pi ก็ถือเป็นรุ่นหลานของเต่า "ช้าและรวนง่ายในระดับหนึ่ง" และที่สำคัญ การใช้ภาษาไทยสำหรับคีย์บอร์ด ยังไม่มีความสมบูรณ์ 100%



บางคนซื้อบอร์ด Raspberry Pi 3 ไปทำเป็น Home Media Center ดูทีวีออนไลน์ ดู YouTube ดูหนังออนไลน์ "โอยยย ชีวิตนี้ไม่ต้องการอะไรอีกแล้ว" แต่เวลาจะดูอะไรที ต้องใช้เมาส์คลิก "เครื่องมือป้อนข้อมูล" บนหน้า Google แล้วใช้เมาส์คลิกเลือกพยัญชนะและสระภาษาไทยทีละตัว เพื่อประสมคำให้ได้ประโยคในสิ่งที่ตัวเองอยากจะดู เพราะระบบปฏิบัติการ Raspbian บนบอร์ด Raspberry Pi ยังไม่มีความสมบูรณ์สำหรับการใช้คีย์บอร์ดภาษาไทย แม้จะมีวิธีการตั้งค่าคีย์บอร์ดภาษาไทยออกมาให้ทำตามมากมายก็ตาม

แต่สำหรับบอร์ด Arduino ข้อดีที่สำคัญสำหรับมือใหม่ก็คือ สามารถเชื่อมโยงกับระบบปฏิบัติการคอมพิวเตอร์ เพราะเป็นโปรแกรมหนึ่งซึ่งทำงานบนระบบปฏิบัติการ หมายถึง Arduino IDE "ไม่ได้แยกออกไปเป็นเอกเทศ

เหมือน Raspbian RPI" เนื่องจากระบบของ Arduino มุ่งการเป็นบอร์ดไมโครคอนโทรลเลอร์เพียงอย่างเดียว ไม่ว่าจะเชื่อมต่อกับคอมพิวเตอร์ PC โน้ตบุ๊ค หรือเชื่อมต่อผ่าน Arduino Web Editor ทั้งหมดสามารถ Upload โปรแกรมชุดคำสั่งไปถึงบอร์ด Arduino ได้อย่างไร้ซึ่งอุปสรรค และที่สำคัญ "สำหรับมือใหม่ใฝ่หาประสบการณ์" จำเป็นที่สุดสำหรับขั้นตอนการ Copy Sketch Code จากผู้รู้ท่านอื่นๆ ที่เผยแพร่ Sketch Code ซึ่งเป็นผลงานของพวกเขาในโลกของอินเตอร์เน็ต เพื่อนำไปวางในโปรแกรม Arduino IDE หรือ Arduino Web Editor ได้อย่างสะดวก ไม่ว่าจะเป็น Copy หรือ Paste หรือ Ctrl + C และ Ctrl + V มันจึงเป็นแรงขับที่ทำให้เราสนุกกับบอร์ด Arduino ได้อย่างไม่รู้จบ

เช่นนั้นในนาทีนี้ หากใครสนใจจะศึกษาหรือลองเล่นบอร์ดไมโครคอนโทรลเลอร์ ขอแนะนำให้ลองเรียนลองเล่นกับบอร์ด Arduino ก่อน อย่าพึ่งกระโดดไปเล่นบอร์ด Raspberry Pi ถ้าสมมุติโจทย์ในใจเรามีเพียงคำว่า "ไมโครคอนโทรลเลอร์" แต่ก็ปฏิเสธไม่ได้ว่าบอร์ด Raspberry Pi มีอะไรที่มากกว่าแค่คำว่า "ไมโครคอนโทรลเลอร์"

แน่นอนถ้าเราตัดสินใจซื้อ 40 IN 1 Sensor Kit for Arduino ไม่ว่าจากค่ายไหนยี่ห้อใดก็ตาม เราจะสามารถสนุกสุดเหวี่ยงกับความสามารถของบอร์ด Arduino ที่ควบคุมคุณสมบัติของชุดโมดูลอิเล็กทรอนิกส์เซ็นเซอร์ได้หลากหลายรูปแบบ และไม่ต้องกังวลกับภาษา C อย่างที่เน้นย้ำไป "อย่าพึ่งหมกมุ่นกับการเขียนโปรแกรม เราจะต้องหัดพูดก่อนที่จะหัดเขียน จำได้มั้ย" สำหรับมือใหม่หัดต่อวงจร Kit for Arduino จงใช้โปรแกรมชุดคำสั่งทั้งหมดที่ได้จากผู้รู้ หรือโปรแกรมชุดคำสั่งตัวอย่างจากหนังสือเล่มนี้ เป็นข้อมูลตัวอย่างเพื่ออ้างอิงในการเรียนรู้การเขียนโปรแกรมด้วยภาษา C ทำไมต้อง else digitalWrite (ledPin, LOW); ทำไมต้อง pinMode (ledPin, OUTPUT) แล้วทำไมต้อง int ledPin = 3; // select the pin for the LED เมื่อสามารถทำให้บอร์ด Arduino ส่งสัญญาณดิจิตอลให้วงจรโมดูลเซ็นเซอร์ทำงานได้ตามคุณสมบัติแล้ว จงหาเหตุผลว่า เพราะเหตุใดจึงต้องใช้คำพวกนี้ ประโยคเหล่านี้เป็นคำสั่งโปรแกรม การกำหนดหมุดของบอร์ด เพื่อให้ส่งสัญญาณไปยังขาอุปกรณ์ "หมุดไหน ไปขา

อุปกรณ์ใด” ตัวอย่างเริ่มต้นใช้คำอะไรในบรรทัดคำสั่งนั้นๆ จงใช้แนวทางปฏิบัติสังเคราะห์ข้อมูลที่เกิดขึ้นใน Sketch Code ซึ่งทำให้เสียงเกิด ไฟ LED ติดหรือดับ หรือประสบความสำเร็จเมื่อทำตามเงื่อนไขที่โมดูลเซ็นเซอร์กำหนดไว้

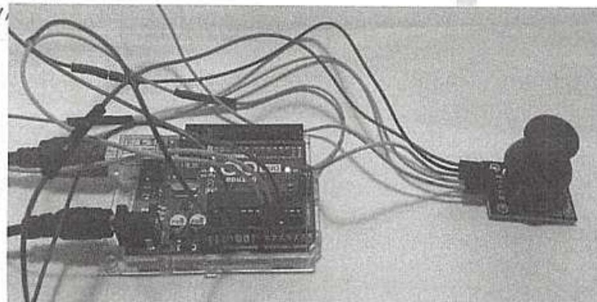
และแน่นอนอีกเช่นกันตัวอย่าง Sketch Code ในหนังสือเล่มนี้ คงไม่ใช่ Code ทั้งหมดของโมดูลเซ็นเซอร์ที่โลกนี้มี มันอาจมากกว่าหมื่นชิ้น “จะต้องมีเซ็นเซอร์บางตัวหรือมากกว่า 10 โมดูล ที่คุณจะต้องเสิร์จหา Sketch Code เอาเองในอินเตอร์เน็ต” เช่นนั้นจงเปิดหัวสมองรับเอาประสบการณ์โปรเจ็กต์ต่อไปนี้ เพื่อต่อยอดพัฒนาประสบการณ์ที่มีต่อบอร์ด Arduino พยายามซึมซับและเพิ่มระดับความผูกพันกับภาษา C บนรูปแบบของโปรแกรม Arduino IDE และ Arduino Web Editor เพื่อให้เราสามารถยืนได้บนลำแข้งของตัวเอง โดยไม่ต้องยืมจมูกคนอื่นหายใจอีกต่อไป ย้ำอีกครั้ง “จงนำแนวทางปฏิบัติไปสังเคราะห์ข้อมูลใน Sketch Code อะไร ที่ไหน เมื่อไหร่” สั่งหมุดนี้ไปขาไหน สั่งหมุดนี้ไปขาอะไร ทำไมถึง HIGH ทำไมถึง LOW... แล้วทำไมต้อง delay(2000);

Analog_Joystick_Test | Arduino 1.8.6 Hourly Build 2018/06/18 03:33

File Edit Sketch Tools Help

Analog_Joystick_Test

```
//
int JoyStick_X = A0; // x
int JoyStick_Y = A1; // y
int JoyStick_Z = A2; // button
void setup ()
{
  pinMode (JoyStick_X, INPUT);
  pinMode (JoyStick_Y, INPUT);
  pinMode (JoyStick_Z, INPUT);
  Serial.begin (9600); //
```



 Temperature Sensor Module

Temperature Sensor คือ โมดูลที่นิยมใช้กันในอุตสาหกรรมที่มีความจำเป็นจะต้องควบคุมอุณหภูมิ หรือห้องเก็บอุปกรณ์สื่อสารแม่ข่ายขั้นสูง ซึ่งจะต้องมีการควบคุมอุณหภูมิให้เหมาะสม โดยในโปรเจ็กต์นี้จะใช้ Temperature Sensor Module รับโปรแกรมชุดคำสั่งจากบอร์ด Arduino ซึ่งผลการปฏิบัติของ Temperature Sensor Module จะส่งสัญญาณค่าอุณหภูมิที่ตรวจวัดได้กลับมายังบอร์ด Arduino และส่งสัญญาณต่อไปยังโปรแกรม Arduino IDE และจะแสดงค่าอุณหภูมิบนมอนิเตอร์ในโปรแกรม Arduino IDE

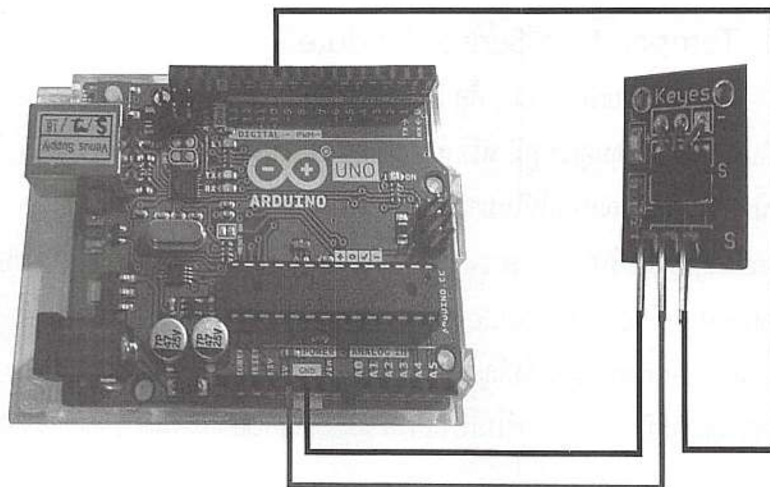
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Temperature Sensor Module
3. Jumper Wire
4. Breadboard



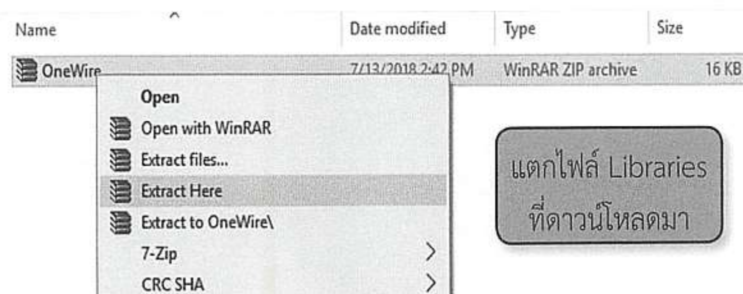
การต่อวงจร

- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา VCC (ขากลาง) ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา S ของเซ็นเซอร์ต่อกับหมุด ~10 บนบอร์ด Arduino



โดยขา S ของวงจรเซ็นเซอร์ จะรับสัญญาณจากตัวเซ็นเซอร์ตรวจวัดอุณหภูมิ และส่งต่อไปยังหมุด ~10 ของบอร์ด Arduino จากนั้นบอร์ดจะส่งสัญญาณดังกล่าวกลับไปยังโปรแกรม Arduino IDE เพื่อทำการประมวลผล และแสดงผลการวัดค่าอุณหภูมิในขณะนั้น

สำหรับการสร้างโปรแกรมชุดคำสั่ง Sketch Code จะต้องใช้ Libraries เฉพาะของเซ็นเซอร์โมดูล โดยให้ไปดาวน์โหลดได้ที่ Link นี้ ***http://osoyoo.com/wp-content/uploads/samplecode/OneWire.zip*** จากนั้นทำการแตกไฟล์ แล้วนำโฟลเดอร์ Libraries ที่ดาวน์โหลดมา นำไปวางไว้ในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE



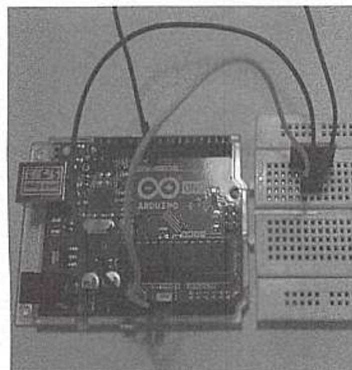
Name	Date modified	Type	Size
OneWire	3/9/2013 6:31 PM	File folder	
OneWire	7/13/2018 2:42 PM	WinRAR ZIP archive	16 KB

โฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

drivers		File f
examples		File f
hardware		File f
java	6/18/2018 3:33 PM	File f
lib	6/18/2018 3:33 PM	File f
libraries	7/12/2018 12:16 AM	File f
reference	6/18/2018 3:33 PM	File f
tools	6/18/2018 3:33 PM	File f
tools-builder	6/18/2018 3:33 PM	File f
arduino	6/18/2018 3:33 PM	Appli
arduino.l4j	6/18/2018 3:33 PM	Confi
arduino_debug	6/18/2018 3:33 PM	Appli

วางโฟลเดอร์ Libraries ใหม่ลงในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

Adafruit_Circuit_Play		
Bridge		
Esplora		
Ethernet		
Firmata		
GSM		
IRremote		
Keyboard	3/8/2016 5:05 PM	File folder
LiquidCrystal	8/10/2017 10:26 AM	File folder
Mouse	3/8/2016 5:04 PM	File folder
OneWire	7/13/2018 2:46 PM	File folder
Robot_Control	8/18/2017 11:29 AM	File folder
Robot_Motor	8/18/2017 12:11 PM	File folder
RobotIRremote	8/18/2017 11:25 AM	File folder
SD	1/5/2017 6:00 PM	File folder
Servo	3/20/2017 5:42 PM	File folder



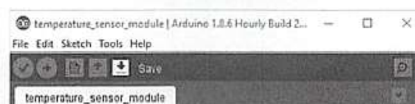
Sketch Code

```
#include <OneWire.h>
int inPin=10; // define D10 as input pin connecting to DS18S20 S pin
OneWire ds(inPin);
void setup(void) {
```

```
 Serial.begin(9600);
}
 void loop(void) {
 int HighByte, LowByte, TReading, SignBit, Tc_100, Whole,
Fract;
 byte i;
 byte present = 0;
 byte Data[12];
 byte addr[8];
 if ( !ds.search(addr)) {
   ds.reset_search();
   return;
 }
 ds.reset();
 ds.select(addr);
 ds.write(0x44,1);
 delay(1000);
 present = ds.reset();
 ds.select(addr);
 ds.write(0xBE);
  for ( i = 0; i < 9; i++) {
  Data[i] = ds.read();
        }
        Serial.print("Temperature: ");
        LowByte = Data[0];
        HighByte = Data[1];
        TReading = (HighByte << 8) + LowByte;
```

```
    SignBit = TReading & 0x8000;
    if (SignBit)
    {
     TReading = (TReading ^ 0xffff) + 1;
    }
    Tc_100 = (6 * TReading) + TReading / 4;
    Whole = Tc_100 / 100;
    Fract = Tc_100 % 100;
    if (SignBit)
    {
      Serial.print(“-”);
    }
    Serial.print(Whole);
    Serial.print(“.”);
    if (Fract < 10)
    {
      Serial.print(“0”);
    }
    Serial.print(Fract);
    Serial.print(“ C\n”);
}
```

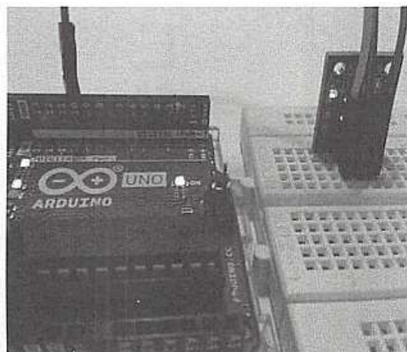
อย่าลืมกด Save



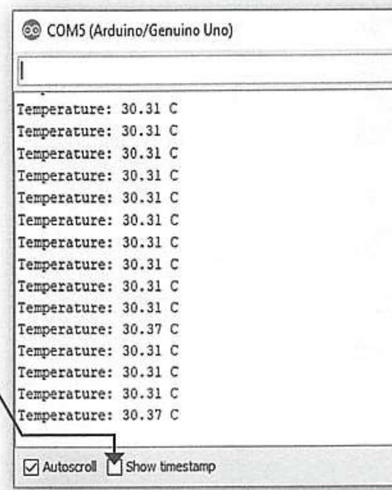
การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Upload เรียบร้อยแล้ว โปรแกรมชุดคำสั่งจะ Run ทันที ให้ไปคลิกที่ Serial Monitor มุมบนด้านขวาของโปรแกรม Arduino IDE



Serial Monitor



คลิกที่ Show timestamp เพื่อให้ระบบแสดงวันและเวลาที่วัดค่าอุณหภูมิแบบ Real Time

COM5 (Arduino/Genuino Uno)

```
Temperature: 30.43 C
Temperature: 30.43 C
Temperature: 30.43 C
Temperature: 30.43 C
Temperature: 30.43 C
Temperature: 30.43 C
14:51:38.174 -> Temperature: 30.43 C
14:51:39.204 -> Temperature: 30.43 C
14:51:40.240 -> Temperature: 30.43 C
14:51:41.264 -> Temperature: 30.43 C
14:51:42.300 -> Temperature: 30.43 C
14:51:43.335 -> Temperature: 30.43 C
14:51:44.373 -> Temperature: 30.43 C
14:51:45.408 -> Temperature: 30.43 C
14:51:46.441 -> Temperature: 30.43 C
```

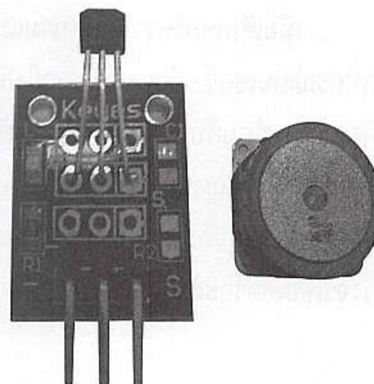
☑ Autoscroll ☑ Show timestamp

 Hall Magnetic Field Sensor

Hall Magnetic Field Sensor คือ เซ็นเซอร์ตรวจจับสนามแม่เหล็ก เมื่อใดที่มีสนามแม่เหล็กเกิดขึ้น ไม่ว่าจะเกิดจากประจุไฟฟ้า หรือเกิดจากโลหะที่มีคุณสมบัติเป็นแม่เหล็ก โมดูลเซ็นเซอร์นี้จะตรวจจับได้ โดยจะใช้ Hall Magnetic Field Sensor รับโปรแกรมคำสั่งจากบอร์ด Arduino ซึ่งสร้างโปรแกรมชุดคำสั่งและ Upload โปรแกรม ผ่านโปรแกรม Arduino IDE

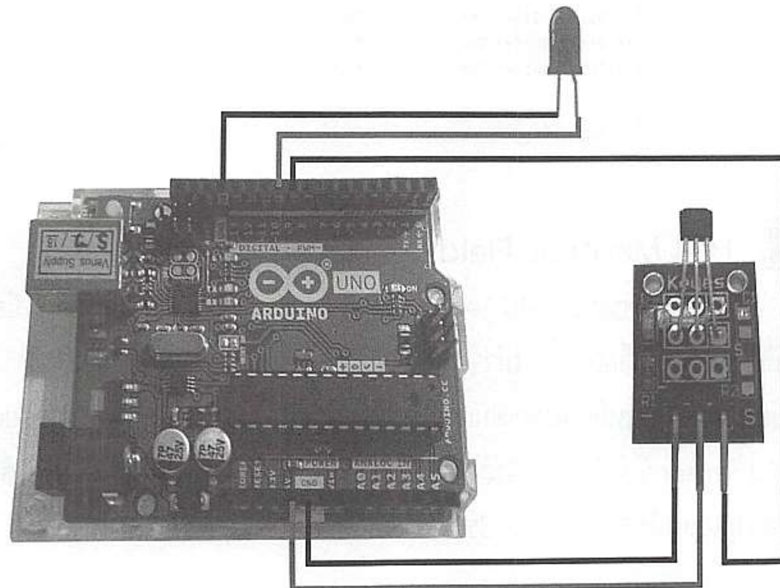
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Hall Magnetic Field Sensor
3. Jumper Wire
4. LED
5. Breadboard
6. แม่เหล็กลำโพง

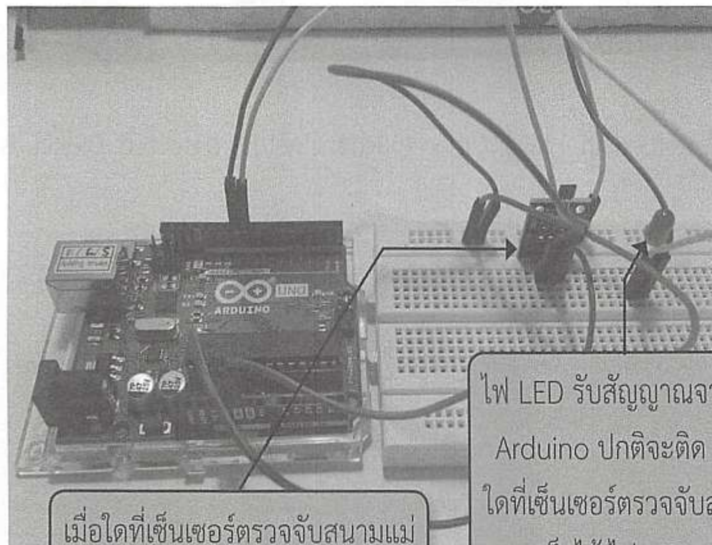


การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~9 บนบอร์ด Arduino
- ขา VCC (ขากลาง) ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด ~10 บนบอร์ด Arduino
- ขา LED - ไป GND ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์ตรวจจับสนามแม่เหล็กได้ จะส่งสัญญาณไปที่ขา S ของอุปกรณ์เซ็นเซอร์ แล้วส่งสัญญาณไปที่ไฟ LED ในวงจรเซ็นเซอร์ ทำให้ไฟ LED ติดสว่าง พร้อมกับส่งสัญญาณต่อไปยังหมุด ~9 ของบอร์ด Arduino เพื่อประมวลผล และตัดสัญญาณไฟ LED ในวงจรผ่านหมุด ~10 ของบอร์ด Arduino ทำให้ไฟ LED ดับ และกลับกัน หากไม่มีอำนาจสนามแม่เหล็กอยู่ใกล้เซ็นเซอร์ ไฟ LED ในวงจรเซ็นเซอร์จะดับ แต่ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติดสว่าง



Sketch Code

```
/* Hall magnetic field sensor test project
 */
int LedPin=10;
int SensorPin=9;
int inputVal;
void setup()
{
pinMode(LedPin,OUTPUT);
pinMode (SensorPin,INPUT);
}
void loop()
{
inputVal=digitalRead(SensorPin);
```

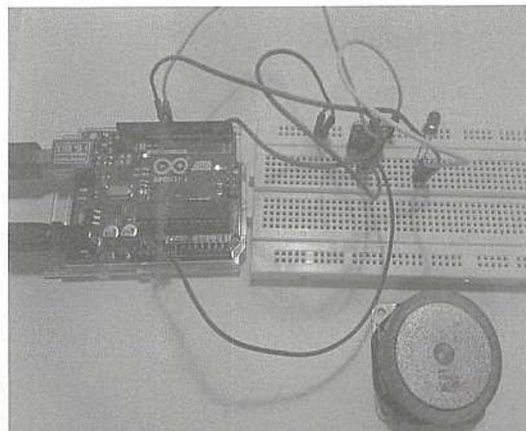
```
if(inputVal==HIGH)
{
 digitalWrite(LedPin, HIGH); //if no magnetic field is detected, D9 LED is off
}
else
{
digitalWrite(LedPin, LOW); //if magnetic field is detected, D9 LED turns on
}
}
```

การ Verify (Compile) และการ Upload

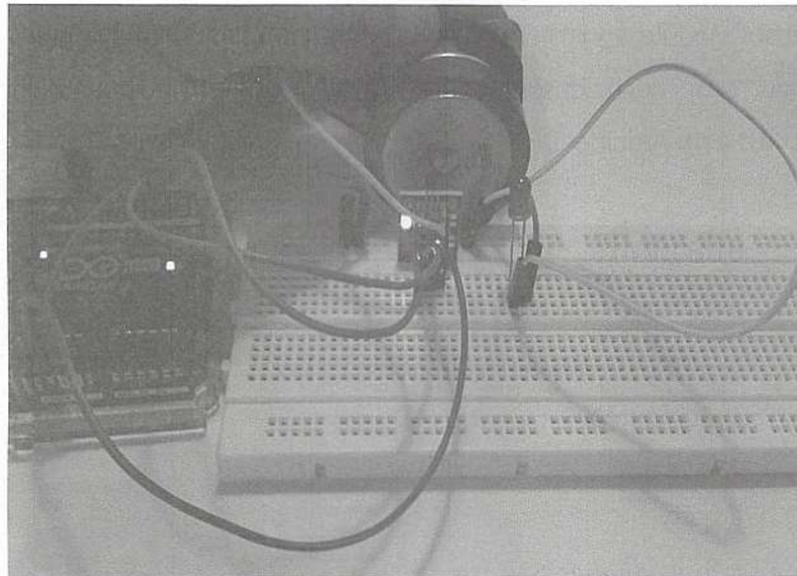


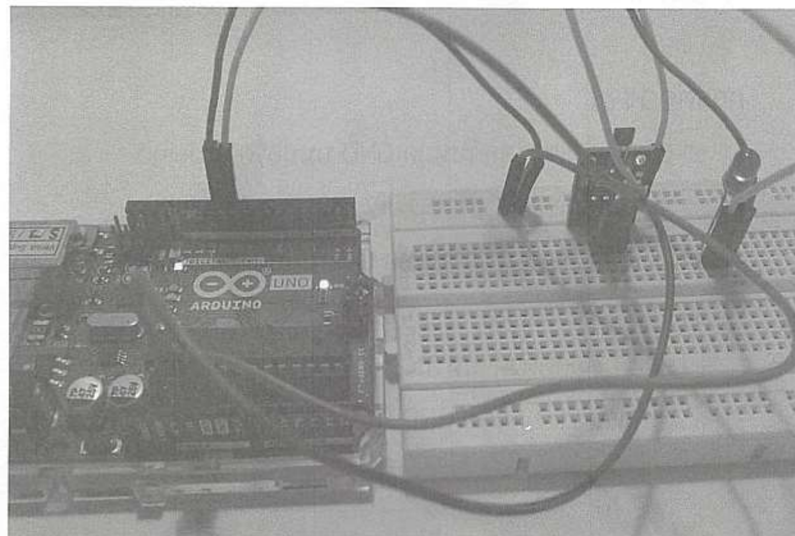


หลังจากที่ระบบทำการ Upload โปรแกรมไปยังบอร์ด Arduino แล้ว วงจรเซ็นเซอร์จะทำงานทันที โดยไฟ LED ของวงจรบอร์ด Arduino จะติด แต่ไฟ LED ในวงจรเซ็นเซอร์จะดับ



เมื่อใดที่เซ็นเซอร์ตรวจพบสนามแม่เหล็ก ไฟ LED ในวงจรเซ็นเซอร์ จะติดทันที แต่ไฟ LED ของวงจรบอร์ด Arduino จะดับลง



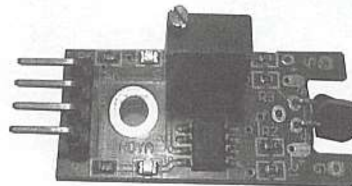


 Touch Sensor Module

Touch Sensor Module คือ อุปกรณ์ที่ใช้ตรวจจับโลหะ โดยนิยมใช้กันอย่างแพร่หลายในวงการอุปกรณ์เซ็นเซอร์กันขโมย ซึ่งคุณสมบัติของ Touch Sensor Module นี้ จะสามารถตรวจจับโลหะทุกชนิดที่มาสัมผัสกับตัวเซ็นเซอร์ ทั้งนี้เราจะให้ Touch Sensor Module รับโปรแกรมชุดคำสั่งจากบอร์ด Arduino ผ่านโปรแกรม Arduino IDE

อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Temperature Sensor Module
3. Jumper Wire
4. ตัวต้านทาน 2kΩ
5. LED
6. Breadboard
7. ไขควงโลหะ

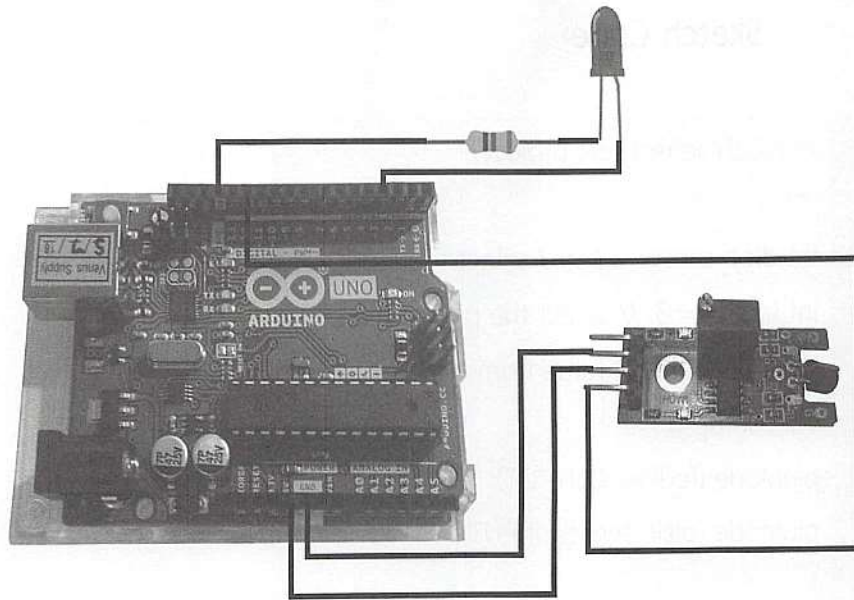


การต่อวงจร

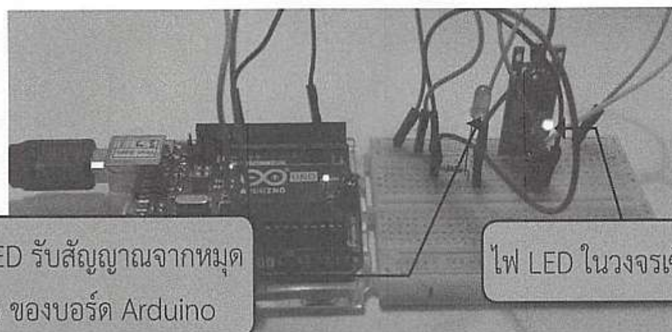
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา DO ของเซ็นเซอร์ต่อกับหมุด D12 บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด ~3 บนบอร์ด Arduino
- ขา LED - ต่อกับตัวต้านทาน 2kΩ
- ตัวต้านทาน 2kΩ ต่อกับ GND บนบอร์ด Arduino



อุปกรณ์ตรวจจับโลหะ



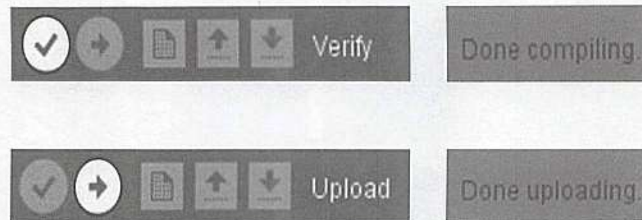
เมื่อโลหะสัมผัสกับเซ็นเซอร์ จะส่งสัญญาณไปที่ไฟ LED ในวงจรเซ็นเซอร์ ทำให้ไฟ LED ติด และพร้อมกันนั้นจะส่งสัญญาณไปที่ขา DO จากนั้นจะส่งสัญญาณไปที่หมุด D12 ของบอร์ด Arduino ทำให้ไฟ LED ที่รับสัญญาณจากหมุด ~3 ติด หากโลหะพ้นจากเซ็นเซอร์แล้ว ไฟ LED ในวงจรเซ็นเซอร์จะดับ และไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะดับพร้อมกันด้วย



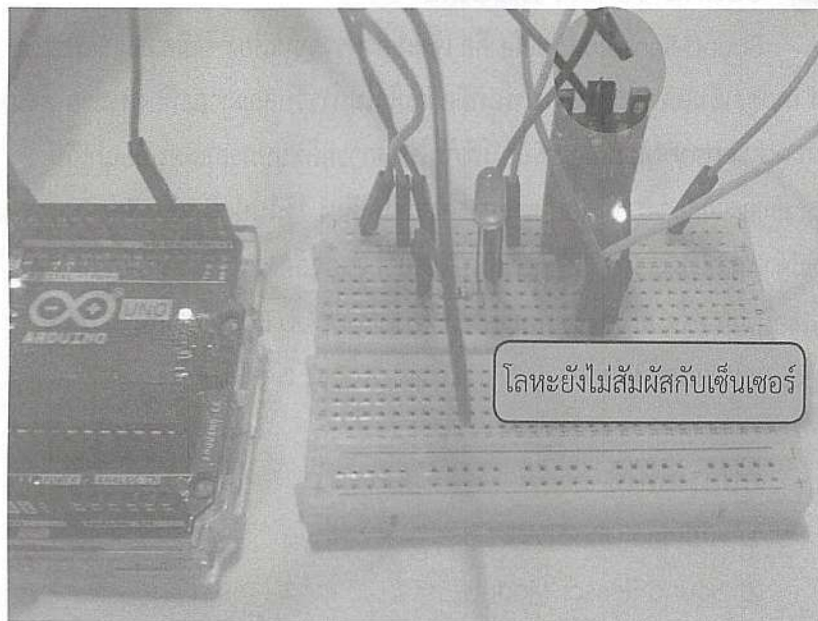
Sketch Code

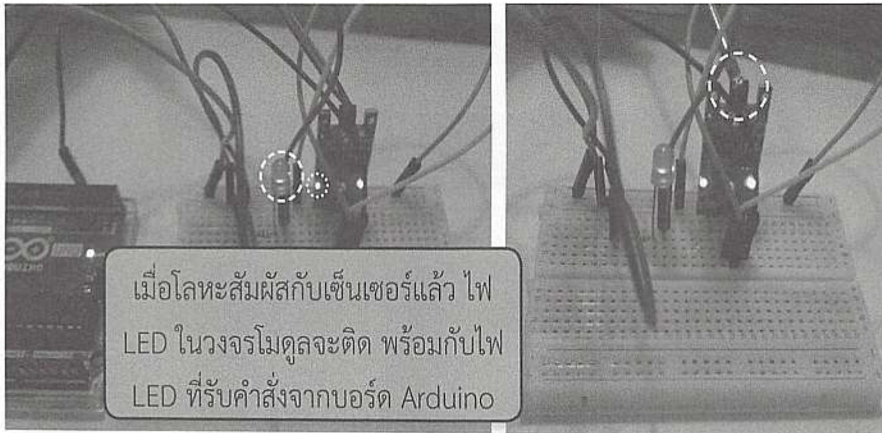
```
/* touch sensor test project
 */
int digit_sensor = 12; // select the input pin for the potentiometer
int ledPin = 3; // select the pin for the LED
int digitValue ; // value from the digit input pin
void setup () {
pinMode (ledPin, OUTPUT);
pinMode (digit_sensor, INPUT);
}
void loop () {
digitValue=digitalRead(digit_sensor);
if (digitValue==HIGH)
 digitalWrite (ledPin, HIGH);
 else digitalWrite (ledPin, LOW);
}
```

การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Upload โปรแกรมเสร็จเรียบร้อยแล้ว บอร์ด Arduino จะส่งสัญญาณผ่านหมุดที่กำหนดไว้ใน Sketch Code ทันที





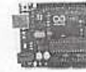
 Sound Sensor Module

Sound Sensor Module คือ เซ็นเซอร์ตรวจจับเสียง ซึ่งมีความ Sensitive สูง โดยชุดเซ็นเซอร์จะมีไมโครโฟนที่มีความไวในการรับสัญญาณเสียงสูงมาก ทั้งนี้ Sound Sensor Module จะนิยมใช้กันในวงการผลิตอุปกรณ์เตือนภัยฉุกเฉินและสัญญาณกันขโมย โดยเฉพาะการตรวจจับเสียง "ที่ไม่ปกติ หรือเสียงที่ปกติแต่มีค่าเกณฑ์ที่ตั้งไว้" ซึ่งเป็นคุณสมบัติเด่นของชุดกล้องวงจรปิดในปัจจุบัน ทั้งนี้จะเชื่อมต่อบอร์ด Arduino กับเว็บไซต์ Arduino Web Editor เพื่อทำการ Upload โปรแกรมชุดคำสั่ง

อุปกรณ์ที่ใช้

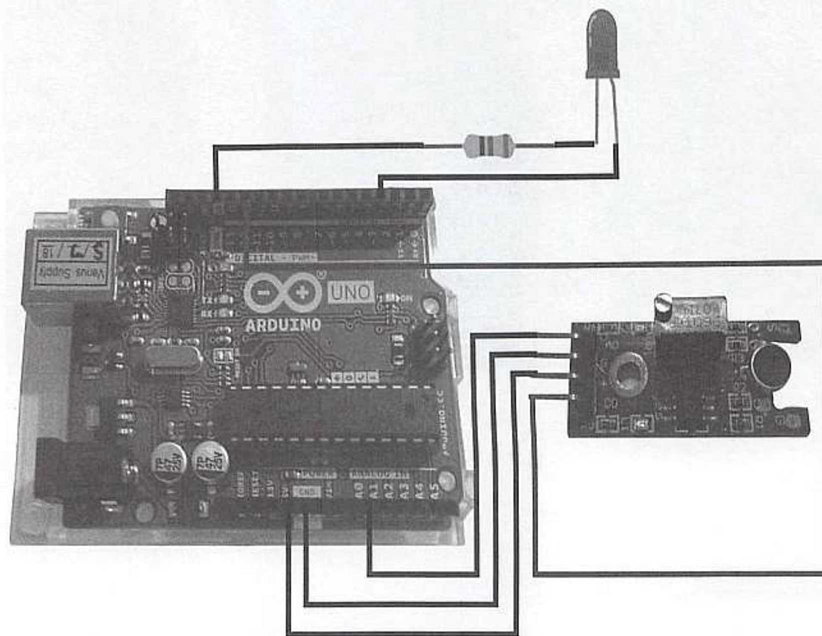
1. Arduino Uno R3
2. Sound Sensor Module
3. Jumper Wire
4. ตัวต้านทาน 2kΩ
5. LED
6. Breadboard

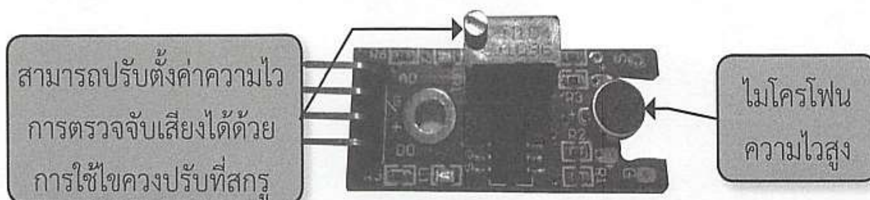




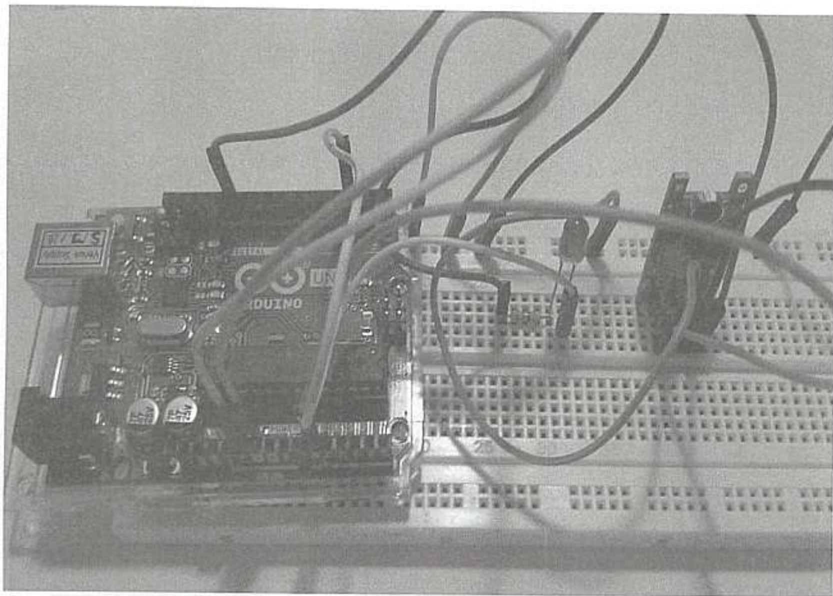
การต่อวงจร

- ขา AO ของเซ็นเซอร์ต่อกับหมุด AO บนบอร์ด Arduino
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา DO ของเซ็นเซอร์ต่อกับหมุด D12 บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด ~3 บนบอร์ด Arduino
- ขา LED - ต่อกับตัวต้านทาน 2kΩ
- ตัวต้านทาน 2kΩ ต่อกับหมุด GND บนบอร์ด Arduino





เมื่อไมโครโฟนสามารถรับสัญญาณเสียงที่เกิดขึ้นได้ จะส่งสัญญาณไปที่บอร์ด Arduino ซึ่งเชื่อมต่อกับโปรแกรม Arduino IDE และทำการประมวลผลค่าเสียงที่ตรวจจับได้ ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติดเมื่อเซ็นเซอร์ตรวจจับสัญญาณเสียงที่ได้ตั้งค่าเกณฑ์ระดับเสียงไว้



Sketch Code

```
/* sound sensor test project
*/
int analog_sensor = A0; // select the input pin for the poten-
tiometer
int digit_sensor = 12; // select the input pin for the potenti-
ometer
int ledPin = 3; // select the pin for the LED
int analogValue ; // value from the analog input pin
int digitValue ; // value from the digit input pin
void setup () {
pinMode (ledPin, OUTPUT);
pinMode (digit_sensor, INPUT);
Serial.begin (9600);
}
void loop () {
analogValue = analogRead (analog_sensor);
digitValue=digitalRead(digit_sensor);
Serial.println (analogValue, DEC);
if (digitValue==HIGH)
      digitalWrite (ledPin, HIGH);
 else digitalWrite (ledPin, LOW);
}
```

sketch_jul14a.ino | ReadMe.adoc

```
/* sound sensor test project
 */
int analog_sensor  = A0; // select the input pin for the potentiometer
int digit_sensor = 12; // select the input pin for the potentiometer
int ledPin = 3; // select the pin for the LED
int analogValue ; // value from the analog input pin
int digitValue ;  //  value from the digit input pin
void setup () {
pinMode (ledPin, OUTPUT);
pinMode (digit_sensor, INPUT);
Serial.begin (9600);
}
void loop () {
analogValue = analogRead (analog_sensor);
digitValue=digitalRead(digit_sensor);
Serial.println (analogValue, DEC);
if (digitValue==HIGH)
  digitalWrite (ledPin, HIGH);
 else digitalWrite (ledPin, LOW);


}
```

อย่าลืม Save และตั้งชื่อไฟล์ Sketch Code

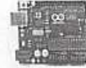
sketch_jul14a

Arduino/Genuino Uno at C... | ••• | SHARE

sketch_jul14a.ino | ReadMe.adoc

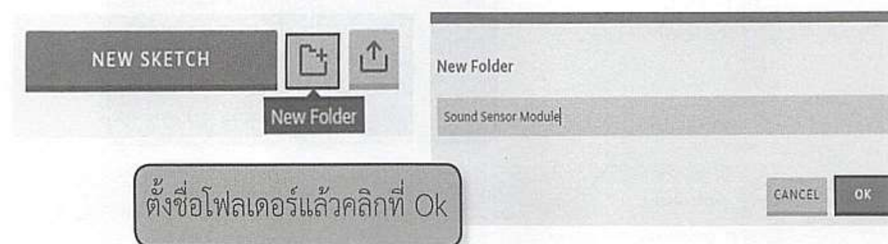
```
/* sound sensor test project
 */
int analog_sensor  = A0; // select the input
int digit_sensor = 12; // select the input p
int ledPin = 3; // select the pin for the LE
int analogValue ; // value from the analog i
int digitValue ;  //  value from the digit i
void setup () {
pinMode (ledPin, OUTPUT);
pinMode (digit_sensor, INPUT);
Serial.begin (9600);
}
```

Save	ctrl S
Save As	shift ctrl S
Rename Sketch...	
Download Sketch	
Export CMake file	
Share Sketch...	
Display Sketch Info...	ctrl I
Delete Sketch...	



การสร้าง Sketch Code ด้วย Arduino Web Editor

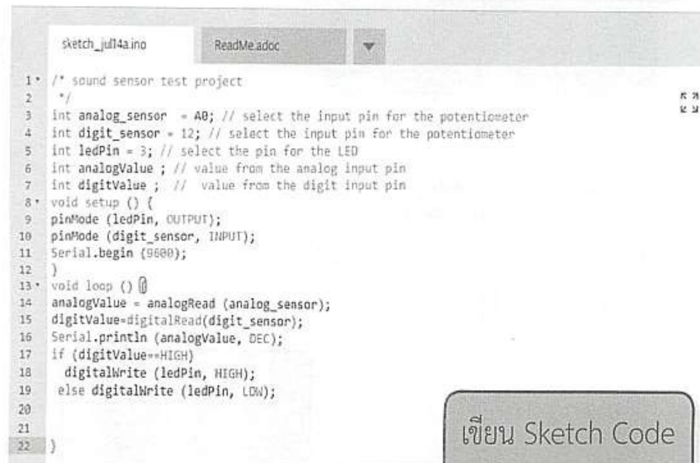
เมื่อเข้าสู่เว็บไซต์ https://create.arduino.cc/editor และ Login เข้าใช้งาน Arduino Web Editor แล้วให้คลิกที่ New Folder เพื่อสร้างโฟลเดอร์สำหรับไฟล์ Sketch Code ของ Sound Sensor Module



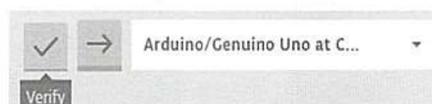


ระบบจะสร้างฟอร์ม Sketch Code ขึ้นมาให้ทันที โดยใน Sketch จะมีคำสั่งเบื้องต้น ซึ่งเราสามารถปรับใช้กับโปรแกรมชุดคำสั่งที่เรากำลังจะสร้างขึ้นใหม่

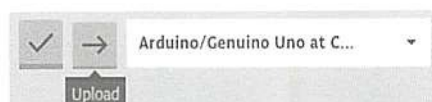




การ Verify (Compile) และการ Upload

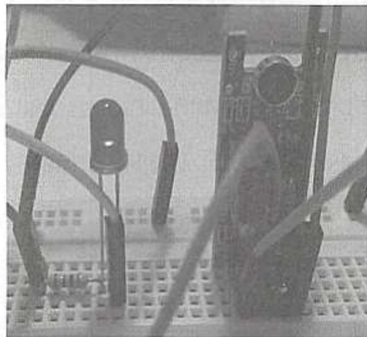


Success: Done verifying



Success: Done uploading

เมื่อระบบ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino เรียบร้อยแล้ว บอร์ด Arduino จะทำงานตามที่โปรแกรมไว้ทันที โดยไมโครโฟนจะตรวจจับเสียงระดับที่เป็นเกณฑ์ซึ่งได้กำหนดค่าไว้ (สกรูที่ชุดเซ็นเซอร์) ซึ่งเซ็นเซอร์จะบันทึกค่าเสียงทั้งหมดตามเกณฑ์ที่เกิดขึ้น และส่งสัญญาณไปที่บอร์ด Arduino จากนั้นจะทำการประมวลผล และแสดงค่าของเสียงที่ Monitor บนเว็บไซต์ Arduino Web Editor โดยให้เราไปที่ถาดเมนูด้านซ้ายมือของเรา แล้วคลิกที่ Monitor



ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติด เมื่อไมโครโฟนสามารถตรวจจับเสียงได้

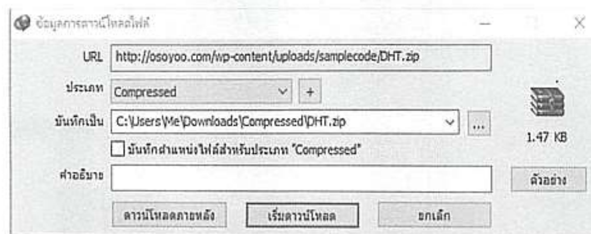
ค่าของเสียงที่ไมโครโฟนรับได้ จะแสดงให้เห็นแบบ Real Time

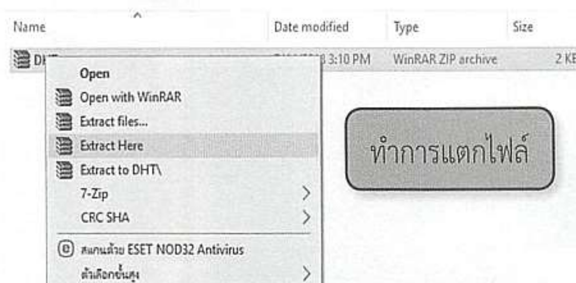
 Temperature Humidity Sensor

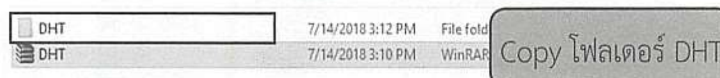
Temperature Humidity Sensor คือ เซ็นเซอร์ตรวจวัดค่าอุณหภูมิและความชื้นในอากาศ โดย Sketch Code ของ Temperature Humidity Sensor จะมี Libraries เฉพาะ ซึ่งเราจะต้องทำการดาวน์โหลดไปเก็บไว้ในโฟลเดอร์ Libraries ของโปรกรม Arduino IDE

การดาวน์โหลดและติดตั้ง Libraries

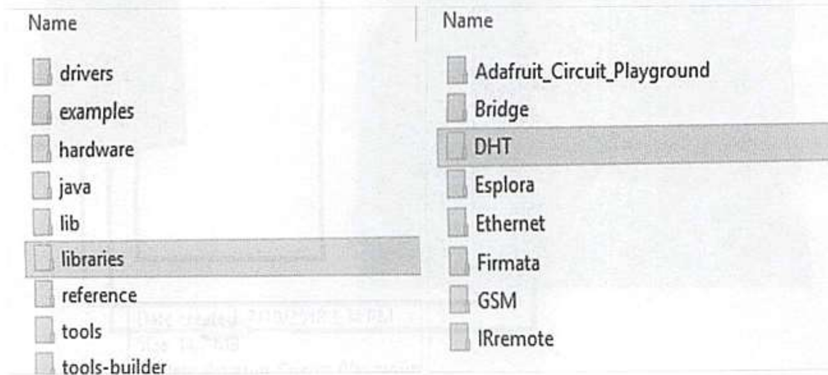
ให้ไปที่ *Link http://osoyoo.com/wp-content/uploads/samplecode/DHT.zip* เมื่อดาวน์โหลด Libraries มาแล้ว ให้ทำการแตกไฟล์ และนำโฟลเดอร์ Libraries ที่มีชื่อ DHT ไปวางในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE



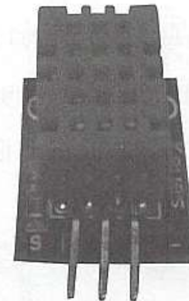




จากนั้นไปที่โฟลเดอร์ Libraries ของโปรแกรม Arduino IDE



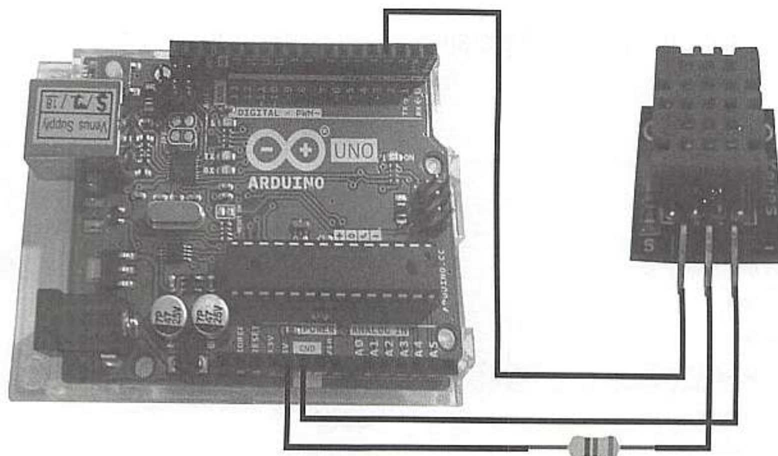
อุปกรณ์ที่ใช้



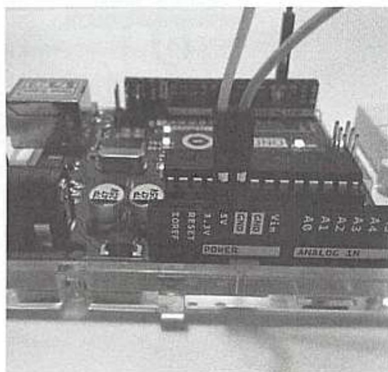
1. Arduino Uno R3
2. Temperature Humidity Sensor
3. Jumper Wire
4. ตัวต้านทาน 2kΩ
5. Breadboard

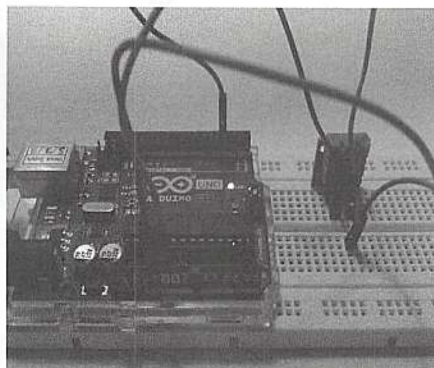
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~3 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับตัวต้านทาน 2kΩ
- ตัวต้านทาน 2kΩ ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์วัดค่าอุณหภูมิและความชื้นในอากาศได้ จะส่งสัญญาณไปที่บอร์ด Arduino ผ่านขา Data ที่ได้เชื่อมต่อกับหมุด ~3 ของบอร์ด Arduino และทำการประมวลผล โดยจะมีการแสดงค่าอุณหภูมิและความชื้นที่ Serial Monitor บนโปรแกรม Arduino IDE





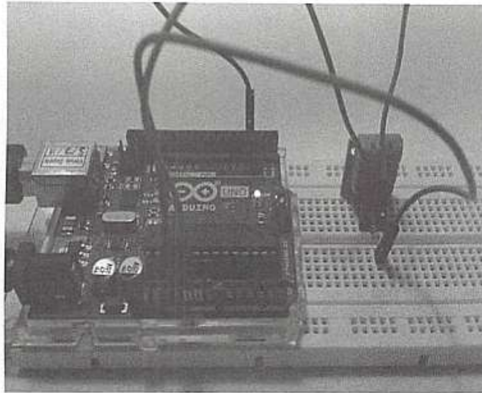
Sketch Code

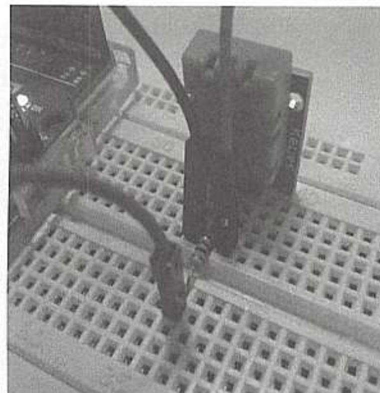
```
/*
 * DHT11 temperature and humidity sensor
 */
#include<dht.h>
dht DHT;
// if you require to change the pin number, Edit the pin with your arduino pin.
#define DHT11_PIN 3
void setup() {
Serial.begin(9600);
Serial.println("Welcome Osoyoo DHT11 Temperature/Humidity
Sample Project"); }
void loop() { // READ Data
int chk = DHT.read11(DHT11_PIN);
Serial.print(" Humidity: " );
Serial.print(DHT.humidity, 1);
Serial.println('%');
Serial.print(" Temparature ");
Serial.print(DHT.temperature, 1);
Serial.println('C');
delay(2000);
}
```

การ Verify (Compile) และการ Upload



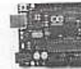
เมื่อระบบทำการ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino เรียบร้อยแล้ว บอร์ดจะส่งสัญญาณและรับสัญญาณค่าอุณหภูมิและค่าความชื้นที่เซ็นเซอร์รับได้ทันที โดยให้เราเปิด Serial Monitor ที่มุมบนขวามือของหน้าต่างโปรแกรม Arduino IDE

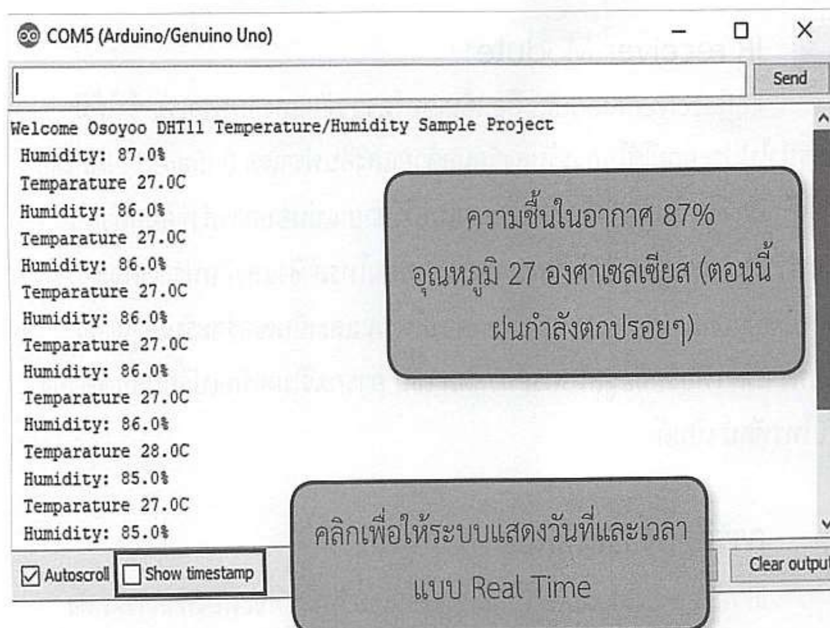


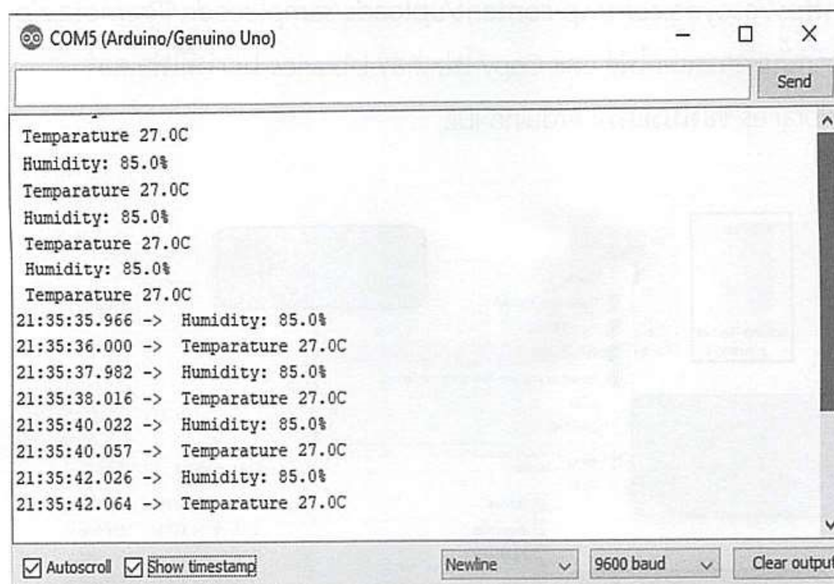




คลิกที่ Serial Monitor




COM5 (Arduino/Genuino Uno)
Send
Welcome Osoyoo DHT11 Temperature/Humidity Sample Project
Humidity: 87.0%
Temparature 27.0C
Humidity: 86.0%
Temparature 27.0C
Humidity: 86.0%
Temparature 27.0C
Humidity: 86.0%
Temparature 27.0C
Humidity: 86.0%
Temparature 27.0C
Humidity: 86.0%
Temparature 28.0C
Humidity: 85.0%
Temparature 27.0C
Humidity: 85.0%
ความชื้นในอากาศ 87%
อุณหภูมิ 27 องศาเซลเซียส (ตอนนี้ฝนกำลังตกปรอยๆ)
คลิกเพื่อให้ระบบแสดงวันที่และเวลาแบบ Real Time
Autoscroll
Show timestamp
Clear output


COM5 (Arduino/Genuino Uno)
Send
Temparature 27.0C
Humidity: 85.0%
Temparature 27.0C
Humidity: 85.0%
Temparature 27.0C
Humidity: 85.0%
Temparature 27.0C
21:35:35.966 -> Humidity: 85.0%
21:35:36.000 -> Temparature 27.0C
21:35:37.982 -> Humidity: 85.0%
21:35:38.016 -> Temparature 27.0C
21:35:40.022 -> Humidity: 85.0%
21:35:40.057 -> Temparature 27.0C
21:35:42.026 -> Humidity: 85.0%
21:35:42.064 -> Temparature 27.0C
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

 IR receiver Module

IR Receiver Module คือ เซ็นเซอร์ตรวจจับแสงอินฟราเรด ซึ่งได้มีการนำไปประยุกต์ใช้ในการรับส่งข้อมูลด้วยแสงอินฟราเรด (Infrared : IR) โดยถือเป็นอีกวิธีหนึ่งในการสื่อสารข้อมูลแบบไร้สาย แน่นอนการส่งข้อมูลด้วยอินฟราเรด ที่เรารู้จักกันดี คือ ระบบรีโมตคอนโทรล ซึ่งแสงอินฟราเรดจะถูกปล่อยออกมาจาก LED ของรีโมตคอนโทรล และเซ็นเซอร์จะรับสัญญาณอินฟราเรด เพื่อรับข้อมูลไปประมวลผล เช่น การกดรีโมตเพื่อเปลี่ยนช่องเครื่องรับโทรทัศน์ เป็นต้น

การติดตั้ง Libraries

IR receiver Module เป็นอีกชุดเซ็นเซอร์หนึ่ง ที่จะต้องทำการติดตั้ง Libraries เพื่อการสร้าง Sketch Code ซึ่งให้เราไปดาวน์โหลด Link จากที่นี่ *http://osoyoo.com/wp-content/uploads/samplecode/IRremote.zip* จากนั้นทำการแตกไฟล์ และ Copy โฟลเดอร์ Libraries ไปวางที่โฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

Arduino-IRremote-master

Open
Open with WinRAR
Extract files...
Extract Here
Extract to Arduino-IRremote-master\
7-Zip
CRC SHA

ทำการแตกไฟล์

Copy โฟลเดอร์ Libraries

จากนั้นไปที่โฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

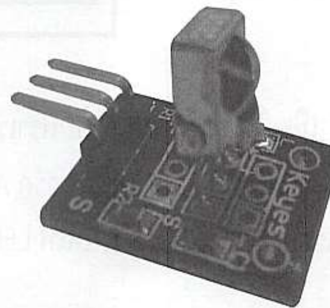
Name	Date modified	Type
drivers	6/18/2018 3:33 PM	File folder
examples	6/18/2018 3:33 PM	File folder
hardware	6/18/2018 3:33 PM	File folder
java	6/18/2018 3:33 PM	File folder
lib	6/18/2018 3:33 PM	File folder
libraries	7/14/2018 3:13 PM	File folder
reference	6/18/2018 3:33 PM	File folder
tools	6/18/2018 3:33 PM	File folder
tools-builder	6/18/2018 3:33 PM	File folder
arduino	6/18/2018 3:33 PM	Application

Name	Date modified	Type
Adafruit_Circuit_Playground	7/18/2017 11:21 PM	File folder
Bridge	8/25/2016 5:20 PM	File folder
DHT	7/14/2018 3:13 PM	File fol
Esplora	5/20/2015 5:10 PM	File fol
Ethernet	6/18/2018 3:33 PM	File fol
Firmata	3/19/2017 12:47 AM	File fol
GSM	6/18/2018 3:33 PM	File fol
IRremote	7/11/2018 11:31 PM	File folder
Keyboard	3/8/2016 5:05 PM	File folder
LiquidCrystal	8/10/2017 10:26 AM	File folder

Paste ลงในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

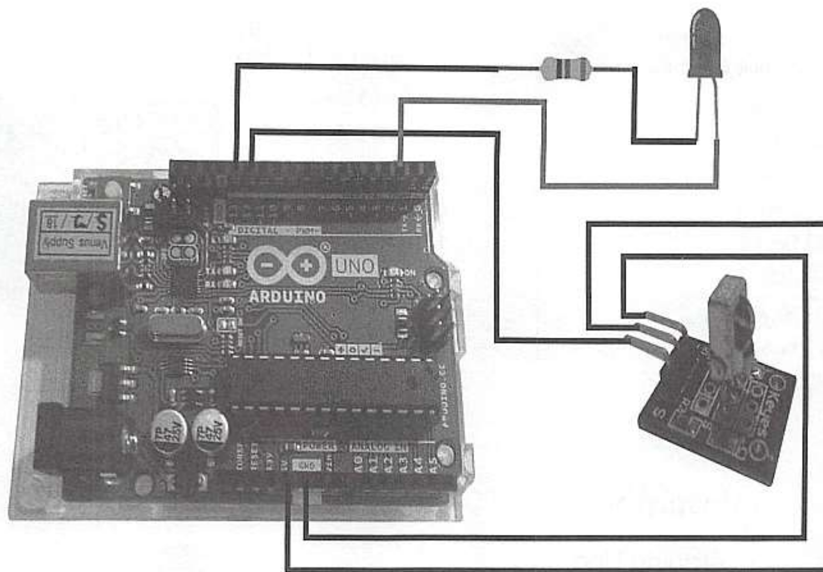
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. IR Receiver Module
3. Jumper Wire
4. LED
5. ตัวต้านทาน 2kΩ
6. Breadboard

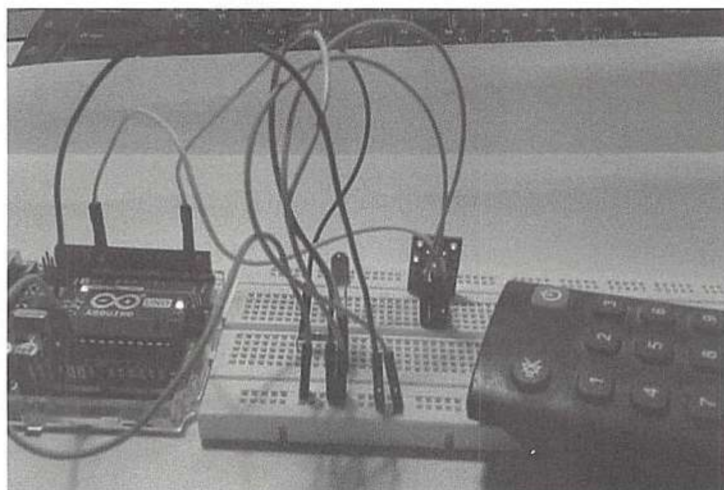


การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด D12 บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด D2 บนบอร์ด Arduino
- ขา LED - ต่อกับตัวต้านทาน 2kΩ
- ตัวต้านทาน 2kΩ ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์ได้รับแสงอินฟราเรด ไฟ LED ในวงจรเซ็นเซอร์จะติด และส่งสัญญาณไปที่หมุด D12 ของบอร์ด Arduino และทำการประมวลผล จากนั้นบอร์ดส่งสัญญาณไปที่ D2 ทำให้ไฟ LED ที่รับสัญญาณจากบอร์ดติดเช่นกัน



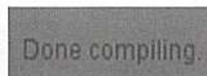
Sketch Code

```
/*
 * IR remote receiver test project
 */
#include <IRremote.h> //adds the library code to the sketch
const int irReceiverPin = 12; //receiver module S pin is connected
to arduino D12
const int ledPin = 2;
IRrecv irrecv(irReceiverPin);
decode_results decodedSignal;
void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  irrecv.enableIRIn();
}
boolean lightState = false;
unsigned long last = millis();
void loop()
{
  if (irrecv.decode(&decodedSignal) == true) //this is true if a mes-
sage has been received
  {
   Serial.println(decodedSignal.value, HEX);
    if (millis() - last > 250) {
```

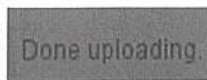
```
      //has it been 1/4 sec since last message
      lightState = !lightState;
      //toggle the LED
      digitalWrite(ledPin, lightState);
    }
    last = millis();
    irrecv.resume();
    // watch out for another message
  }
}
```

การ Verify (Compile) และการ Upload

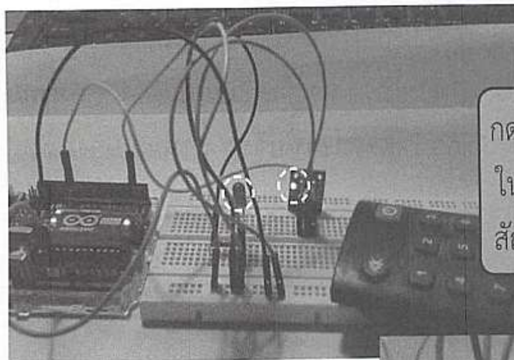




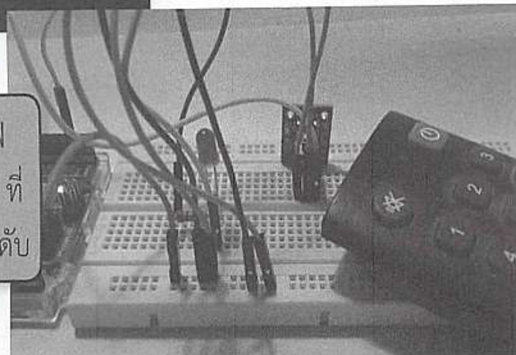




เมื่อระบบทำการ Upload Sketch Code ไปยังบอร์ด Arduino เรียบร้อยแล้ว บอร์ดจะ Run ทันที โดยในการทดลองเราจะใช้รีโมตของเครื่องโทรทัศน์ เพื่อส่งแสงอินฟราเรดไปยังเซ็นเซอร์



กดปุ่มใดๆ ของรีโมต 1 ครั้ง ไฟ LED ในวงจรเซ็นเซอร์ และไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติด



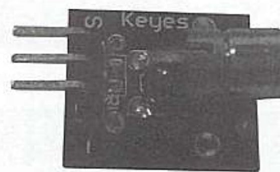
กดปุ่มใดๆ ของรีโมตอีก 1 ครั้ง ไฟ LED ในวงจรเซ็นเซอร์ และไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะดับ

 Laser Module

Laser Module คือ อุปกรณ์ส่งแสงเลเซอร์ ซึ่งมีการนำ Laser Module ไปใช้กับตัวต้านทานไวแสง หรือ LDR สำหรับวงจรการตรวจจับความเคลื่อนไหวในพื้นที่ต้องห้าม หรือผ่านเข้าพื้นที่ในยามวิกาล โดยเมื่อใดที่มีการปิดกั้นลำแสงเลเซอร์ไม่ให้ผ่านไปยังตัวต้านทานไวแสง (เดินผ่านแสงเลเซอร์) ตัวต้านทานไวแสงจะส่งสัญญาณข้อมูลไปยังบอร์ดไมโครคอนโทรลเลอร์ทันที เช่นนั้นเรามาลองเชื่อมต่อ Laser Module กับบอร์ด Arduino ให้ติดสว่างเป็นจังหวะ เพื่อเป็นการเพิ่มประสบการณ์ที่ดียิ่งขึ้น

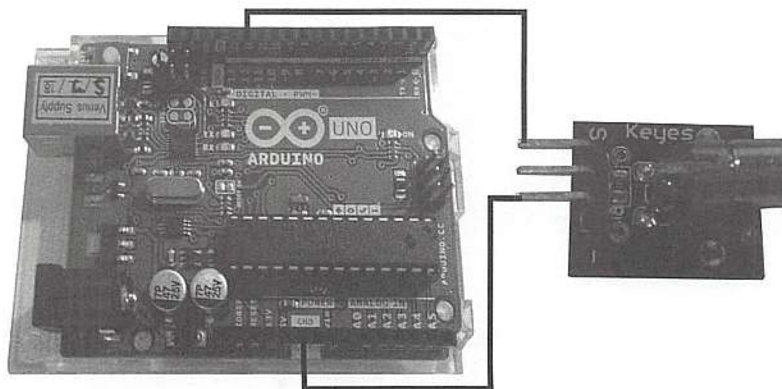
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Laser Module
3. Jumper Wire
4. Breadboard

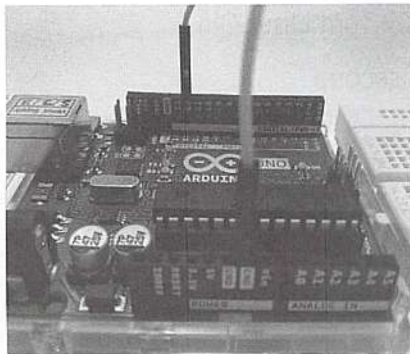


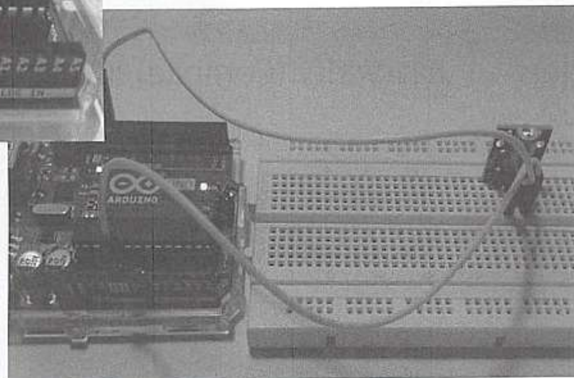
การต่อวงจร

- ขา S ของอุปกรณ์ต่อกับหมุด D12 บนบอร์ด Arduino
- ขา - ของอุปกรณ์ต่อกับหมุด GND บนบอร์ด Arduino



โดย Sketch Code จะกำหนดให้หมุด D12 ส่งสัญญาณไปที่ Laser Module และกำหนดให้ Laser ติด และดับเป็นจังหวะ





Sketch Code

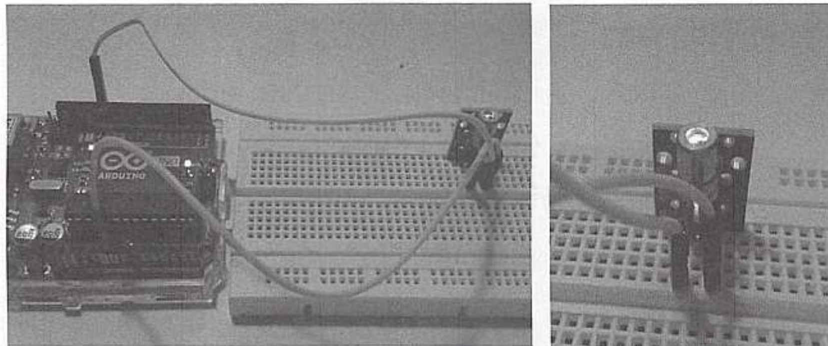
```
// KY-008 Tutorial... On then Off
int laserPin = 12;
void setup ()
{
  pinMode (laserPin, OUTPUT); // define the digital output
interface 12 feet
}
```

```
void loop () {
  digitalWrite (laserPin, HIGH); // Turn Laser On
  delay (1000); // On For Half a Second
  digitalWrite (laserPin, LOW); // Turn Laser Off
  delay (250); // Off for half a second
}
```

การ Verify (Compile) และการ Upload



หลังจากที่ระบบทำการ Upload เสร็จเรียบร้อยแล้ว บอร์ด Arduino จะส่งสัญญาณตามโปรแกรมทันที โดย Laser Module จะติดและดับเป็นจังหวะ ติด 1,000 มิลลิวินาที หน่วงเวลา 250 มิลลิวินาที

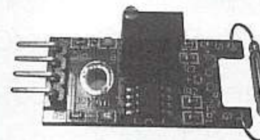


 Reed switch

Reed Switch คือ เซ็นเซอร์สวิตช์แม่เหล็กซึ่งได้มีการนำมาใช้งานกันอย่างแพร่หลาย ไม่ว่าจะเป็นวงการเครื่องยนต์หรือวงการอิเล็กทรอนิกส์ โดยการทำงานของ Reed Switch จะใช้หลักการเหนี่ยวนำของแม่เหล็ก ทั้งนี้ Reed Switch จะมีลักษณะปกติปิดวงจร (Swith Off) แต่ถ้าเมื่อใดที่ได้รับอำนาจของสนามแม่เหล็ก Reed Switch จะเปิดวงจร (Swith On) โดยโปรเจ็กต์นี้เราจะทดลองนำ Reed Switch มารับคำสั่งจากบอร์ด Arduino ตามโปรแกรม Sketch Code

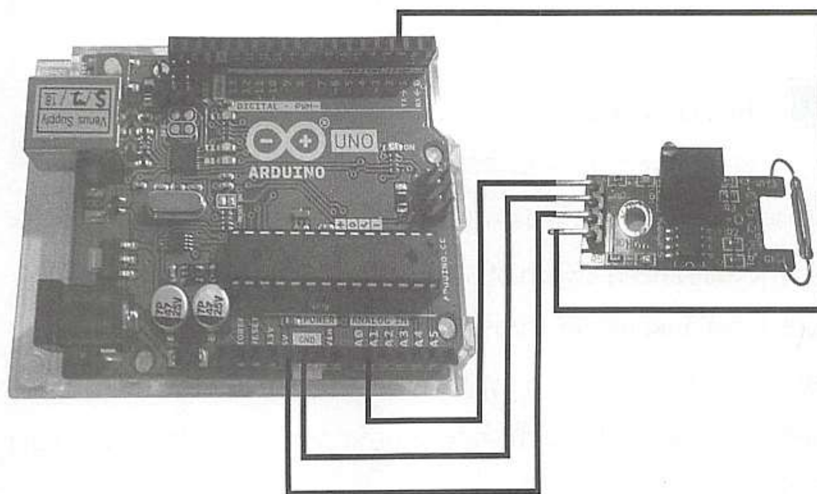
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Reed Switch Sensor Module
3. Jumper Wire
4. Breadboard
5. แม่เหล็กลำโพง

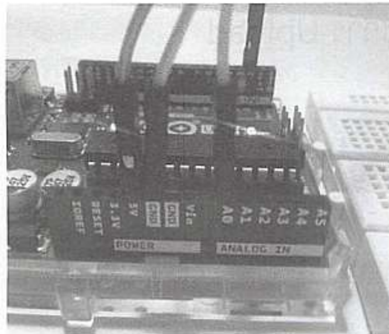


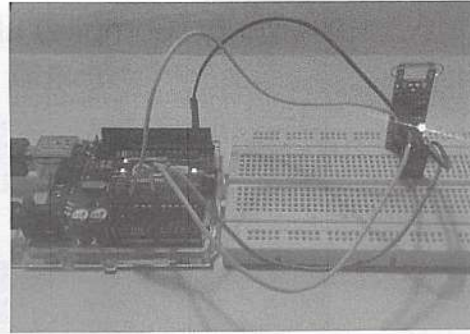
การต่อวงจร

- ขา AO ของเซ็นเซอร์ต่อกับหมุด AO บนบอร์ด Arduino
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา DO ของเซ็นเซอร์ต่อกับหมุด D2 บนบอร์ด Arduino



เมื่อ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว Reed Switch Sensor Module จะมีลักษณะปิดวงจร หมายถึง หน้าสัมผัสแยกออกจากกัน (Off) แต่ถ้าเมื่อใดที่เซ็นเซอร์อยู่ภายในสนามแม่เหล็ก หน้าสัมผัสจะเลื่อนเข้ามาแตะกันเป็นลักษณะเปิดวงจร (On) ไฟ LED ในวงจรเซ็นเซอร์จะเปล่งแสงเพื่อแสดงสถานะ On โดยในตัวอย่างจะใช้แม่เหล็กลำโพงสร้างสนามแม่เหล็กใกล้เซ็นเซอร์ ทั้งนี้บอร์ด Arduino จะได้รับสัญญาณอะนาล็อกจากเซ็นเซอร์ และทำการประมวลผล จากนั้นจะแสดงค่าอะนาล็อกใน Serial Monitor





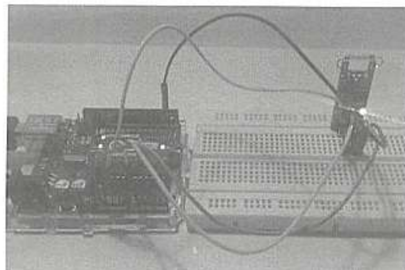
Sketch Code

```
// Arduino pin numbers
//D2 and A0 used
const int digital = 2;
const int analog = 0;
void setup()
{
pinMode(digital, INPUT);
Serial.begin(115200);
}
void loop()
{
Serial.print(digitalRead(digital));
Serial.print("-");
Serial.println(analogRead(analog));
delay(250);
}
```

การ Verify (Compile) และการ Upload



หลังจากที่ระบบทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว บอร์ดจะส่งสัญญาณไปยัง Reed Switch Sensor Module โดยเซ็นเซอร์จะอยู่ในตำแหน่ง Off เมื่อใดที่เกิดสนามแม่เหล็ก เซ็นเซอร์จะเปลี่ยนสถานะไปอยู่ในตำแหน่ง On ทันที ซึ่งไฟ LED แสดงสถานะ On จะเปล่งแสงขึ้น จนกว่าสนามแม่เหล็กจะหมดไป



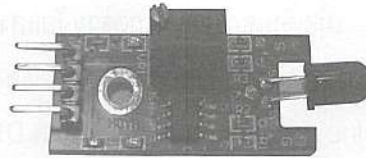


 Flame Sensor Module

Flame Sensor Module คือ เซ็นเซอร์ตรวจจับเปลวไฟด้วยแสงอินฟราเรด ซึ่งใช้กันอย่างแพร่หลายในลักษณะงานด้านความปลอดภัย ไม่ว่าจะเป็นอาคาร สำนักงาน ตึกที่พักอาศัย บ้านเรือน หรือห้องเก็บอุปกรณ์สื่อสาร ฯลฯ ซึ่งอุปกรณ์เซ็นเซอร์ดังกล่าว จะส่งสัญญาณไปยังอุปกรณ์เตือนภัยของสถานที่ทันที เมื่อตรวจจับเปลวไฟที่เกิดจากการเกิดเหตุเพลิงไหม้ โดย Flame Sensor Module จะส่งสัญญาณเมื่อตรวจจับแสงอินฟราเรดที่เกิดจากเปลวไฟได้ ทั้งนี้เราจะนำ Flame Sensor Module ไปทำงานร่วมกับบอร์ด Arduino

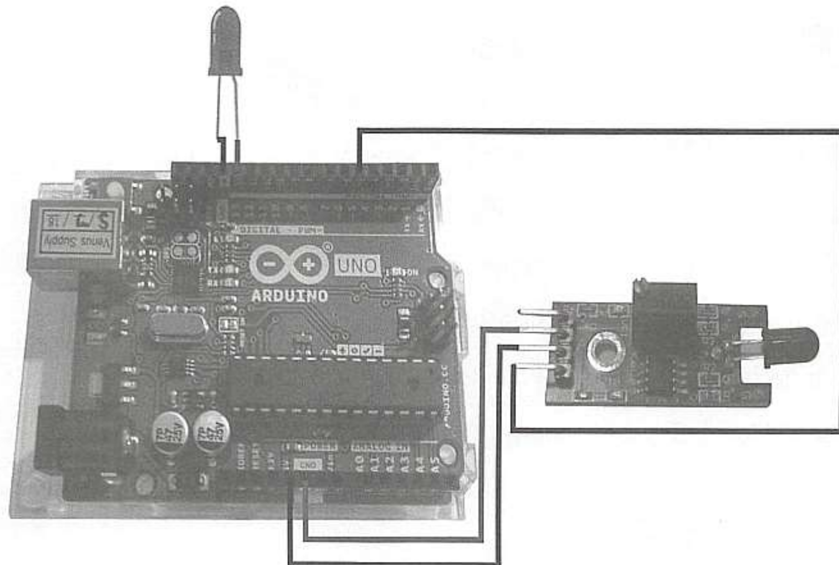
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Flame Sensor Module
3. Jumper Wire
4. Breadboard
5. ไฟแช็ก



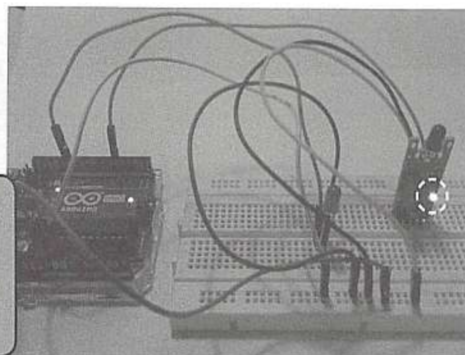
การต่อวงจร

- ขา DO ของเซ็นเซอร์ต่อกับหมุด ~5 บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ของเซ็นเซอร์ต่อกับหมุด D12 บนบอร์ด Arduino
- ขา LED - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์สามารถตรวจจับแสงอินฟราเรดที่เกิดจากเปลวไฟได้ ไฟ LED ในวงจรเซ็นเซอร์จะเปล่งแสง และส่งสัญญาณไปที่หมุด ~5 จากนั้นบอร์ด Arduino จะส่งสัญญาณออกจากหมุด D12 ไฟ LED จะเปล่งแสง





เมื่อเชื่อมต่อกับบอร์ด Arduino ไฟ LED On จะเปล่งแสงแสดงสถานะพร้อมใช้งาน

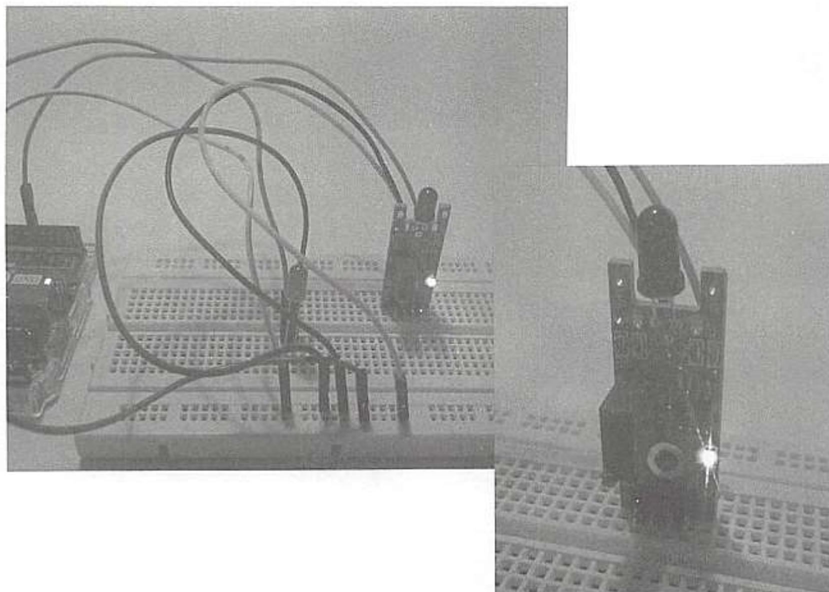
Sketch Code

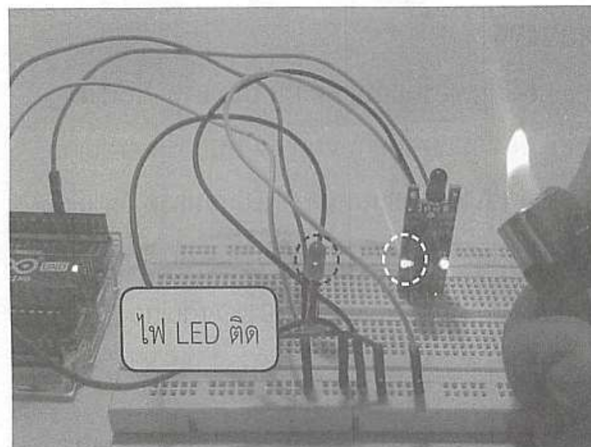
```
*/
int Led=12;
int buttonpin=5;
int val;
void setup()
{
pinMode(Led,OUTPUT);
pinMode(buttonpin,INPUT);
}
void loop()
{
val=digitalRead(buttonpin);
if(val==HIGH)
{
digitalWrite(Led,HIGH);
}
else
{
digitalWrite(Led,LOW);
}
}
```

การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Sketch Code ไปยังบอร์ด Arduino โปรแกรมชุดคำสั่งจะ Run ทันที เมื่อนำไฟแช็กไปทำให้เกิดเปลวไฟในระยะ 10 เซนติเมตร โดยรอบเซ็นเซอร์ ไฟ LED ในวงจรเซ็นเซอร์จะติด และไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติดเช่นกัน เพราะเซ็นเซอร์สามารถตรวจจับแสงอินฟราเรดซึ่งเป็นรังสีความร้อนได้





 Tracking Sensor

Tracking Sensor คือ เซ็นเซอร์ที่สามารถตรวจจับพื้นหรือเส้นสีขาวและสีดำ โดยมีการนำไปใช้กับ Smart Car หรือหุ่นยนต์ เพื่อหลบหลีกหรือเดินตามเส้น ทั้งนี้เซ็นเซอร์จะทำงานโดยใช้หลักการตรวจจับแสงอินฟราเรด ที่สะท้อนกลับมาจากพื้นหรือเส้นที่มีสีขาวและสีดำ เช่น แสงอินฟราเรดจะสะท้อนกลับหากกระทบพื้นหรือเส้นที่เป็นสีขาว แต่หากเป็นพื้นหรือเส้นที่เป็นสีดำ แสงอินฟราเรดจะถูกดูดกลืน แสงอินฟราเรดจึงไม่สามารถสะท้อนกลับได้

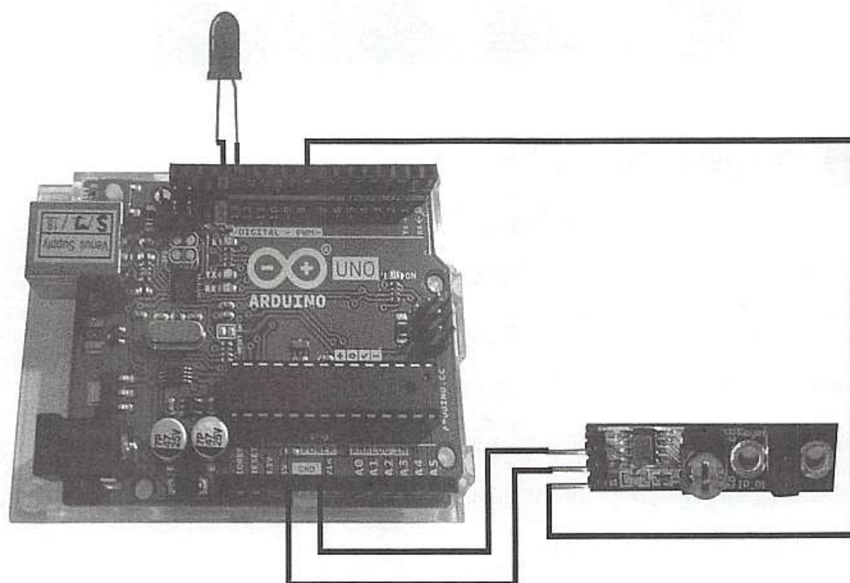
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Tracking Sensor Module
3. Jumper Wire
4. LED
5. กระดาษสีขาว และฝนด้วยดินสอดำ

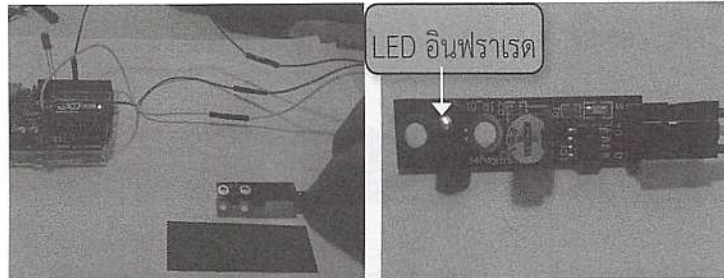


การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด D8 บนบอร์ด Arduino
- ขา V+ ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด D13 บนบอร์ด Arduino
- ขา LED - ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อหลอดอินฟราเรดลงพื้นสีขาว แสงอินฟราเรดจะสะท้อนกลับ ตัวรับจะตรวจจับแสงอินฟราเรดได้ และส่งสัญญาณไปขา S ซึ่งเป็นสัญญาณต่ำ ไฟ LED ในวงจรเซ็นเซอร์จะติด แต่ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะดับ แต่ถ้าวางหลอดอินฟราเรดลงบนพื้นสีดำ แสงอินฟราเรดจะถูกดูดกลืน จึงไม่สามารถสะท้อนกลับได้ เมื่อตัวรับไม่สามารถตรวจจับแสงอินฟราเรดได้ ไฟ LED ในวงจรจึงดับ แต่ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติด



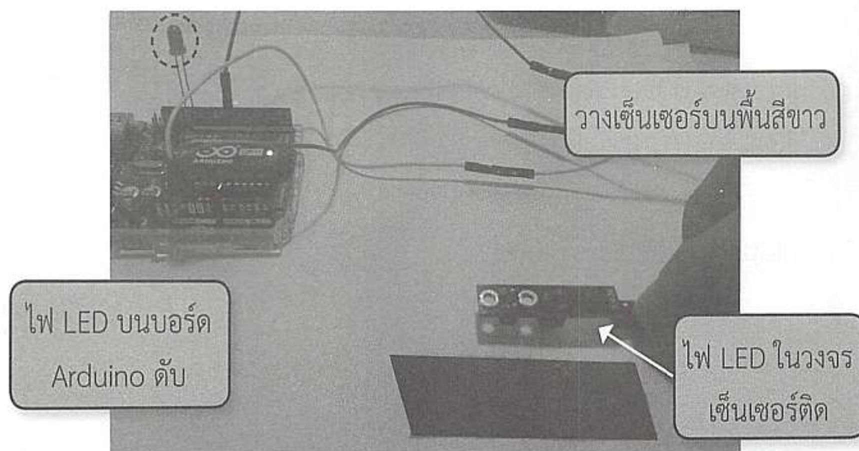
Sketch Code

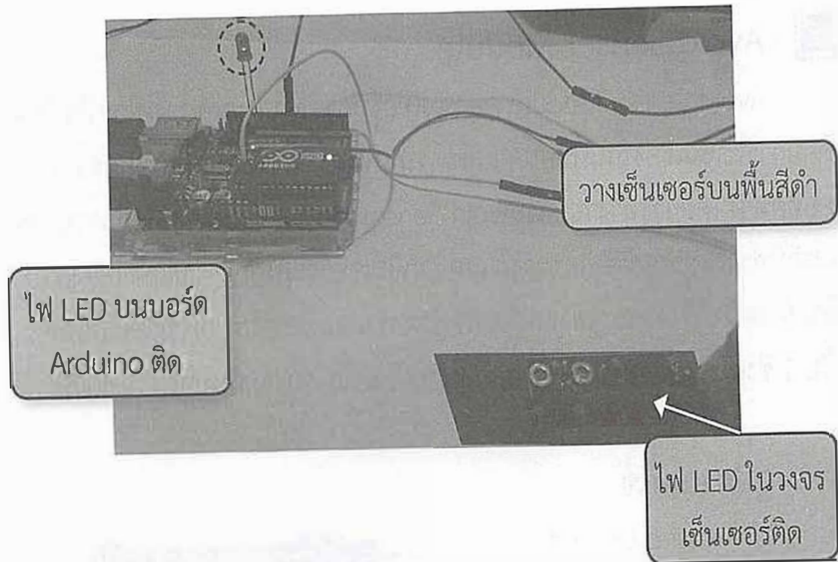
```
const int tracingPin = 8;
const int ledPin = 13;
void setup()
{
pinMode(tracingPin, INPUT);
pinMode(ledPin, OUTPUT);
}
void loop()
{
int val = digitalRead(tracingPin);
if(val == HIGH)
{
digitalWrite(ledPin, HIGH);
}
else
{
digitalWrite(ledPin, LOW);
}
}
```

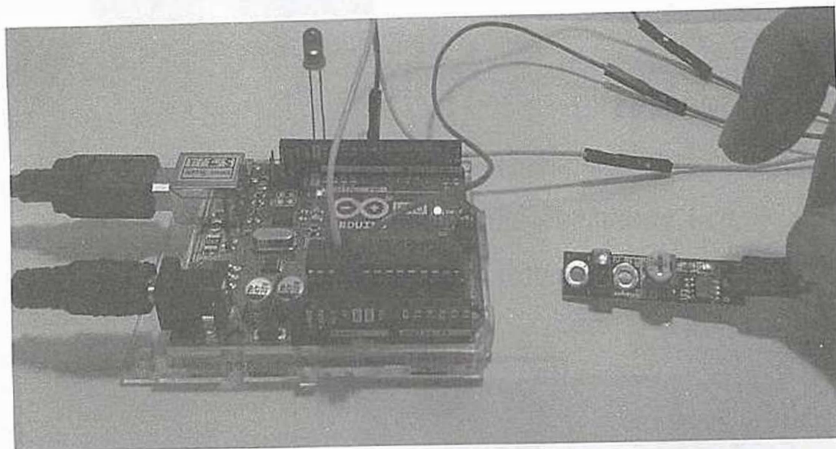
การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Upload Sketch Code เสร็จเรียบร้อยแล้ว โปรแกรมจะ Run ทันที เมื่อวางหลอดอินฟราเรดลงบนพื้นสีขาว แสงอินฟราเรดจะสะท้อนกลับ จนเซ็นเซอร์ตรวจจับได้ ไฟ LED ในวงจรเซ็นเซอร์จะติด แต่ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะดับ แต่หากวางหลอดอินฟราเรดลงบนพื้นสีดำ แสงอินฟราเรดจะถูกสีดำดูดกลืนไม่สามารถสะท้อนกลับได้ เมื่อเซ็นเซอร์ไม่สามารถตรวจจับแสงอินฟราเรดได้ ไฟ LED ในวงจรเซ็นเซอร์จะดับ แต่ไฟ LED ที่รับสัญญาณจากบอร์ด Arduino จะติด




วางเซ็นเซอร์บนพื้นสีดำ
ไฟ LED บนบอร์ด Arduino ติด
ไฟ LED ในวงจรเซ็นเซอร์ติด

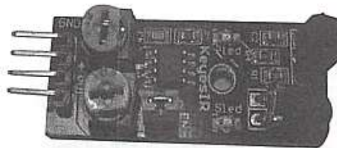


 Avoid Sensor Module

Avoid Sensor Module คือ เซ็นเซอร์ที่ตรวจจับสิ่งกีดขวางเพื่อหลีกเลี่ยง โดยมีการใช้เซ็นเซอร์นี้ในหุ่นยนต์ และโดรน เพื่อใช้ในการหลบหลีกสิ่งกีดขวางโดยทิศทาง โดยการทำงานของ Avoid Sensor Module จะอาศัยหลักการปล่อยแสงอินฟราเรด หากมีสิ่งกีดขวางด้านหน้า เซ็นเซอร์จะส่งสัญญาณและแสดงค่าเป็น 0 แต่ถ้ามีสิ่งกีดขวางลำแสงอินฟราเรด เซ็นเซอร์จะส่งสัญญาณและแสดงค่าเป็น 1 ซึ่งเราจะนำ Avoid Sensor Module มาใช้งานร่วมกับบอร์ด Arduino

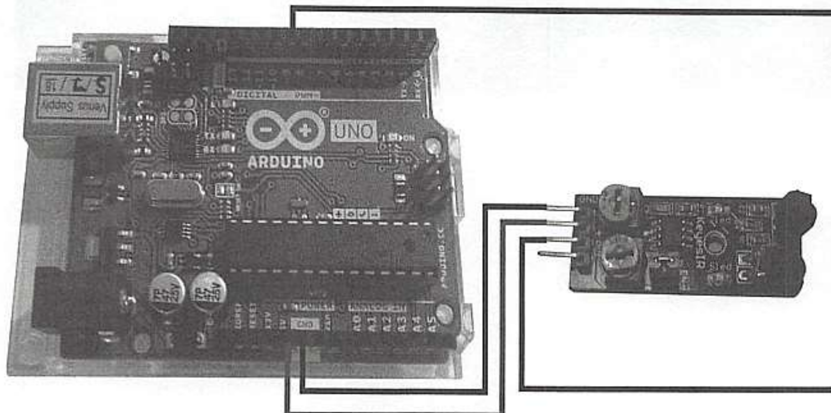
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Avoid Sensor Module
3. Jumper Wire

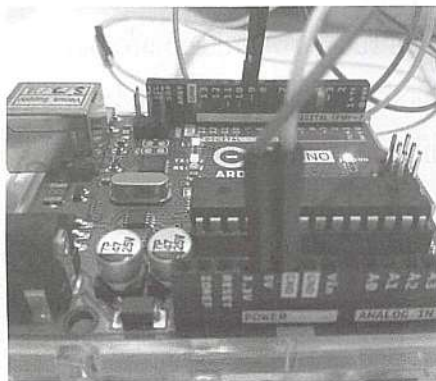


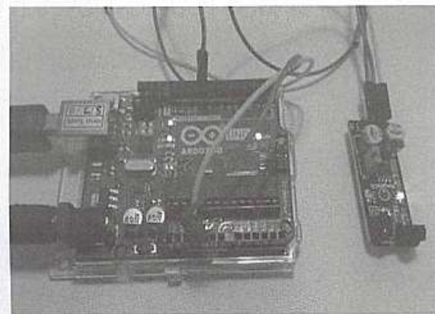
การต่อวงจร

- ขา GND ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา OUT ของเซ็นเซอร์ต่อกับหมุด ~9 บนบอร์ด Arduino



เมื่อแสงอินฟราเรดถูกปล่อยออกไป หากไม่กระทบกับสิ่งกีดขวาง จะส่งสัญญาณไปที่หมุด ~9 แสดงค่าเป็น 0 แต่ถ้าแสงอินฟราเรดมีสิ่งกีดขวาง จะส่งสัญญาณและแสดงค่าเป็น 1 และไฟ LED ในวงจรเซ็นเซอร์จะติด โดยเราสามารถดูค่าได้ที่ Serial Monitor ที่มุมบนขวาของหน้าต่างโปรแกรม Arduino IDE





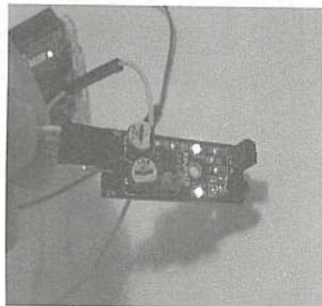
Sketch Code

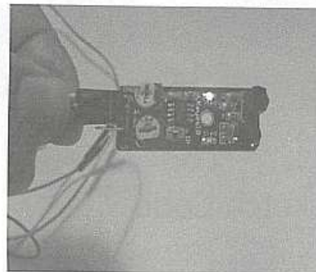
```
*/
int count;
void setup() {
 Serial.begin (9600);
 pinMode (9, INPUT); //Sensor output
}
void loop() {
Serial.print (“Sensor: “);
 Serial.println (digitalRead(9)); //print the sensor output
 delay (500); //wait half a second
}
```

การ Verify (Compile) และการ Upload

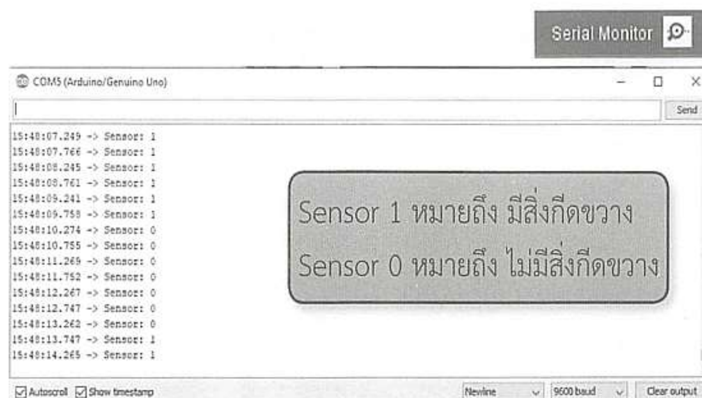


เมื่อทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว วงจรเซ็นเซอร์จะทำงานตามโปรแกรมทันที เมื่อนำเซ็นเซอร์โมดูลชี้ลงที่พื้น ไฟ LED ในวงจรจะติด แต่หากยกเซ็นเซอร์ขึ้นซึ่งไม่มีสิ่งกีดขวางแสงอินฟราเรด ไฟ LED ในวงจรเซ็นเซอร์จะดับ





เราสามารถดูค่าตัวเลขที่เซ็นเซอร์โมดูลส่งสัญญาณไปที่บอร์ด Arduino และทำการประมวลผลเป็นตัวเลข โดยให้ไปที่ Serial Monitor ที่มุมบนด้านขวาของหน้าต่างโปรแกรม Arduino IDE

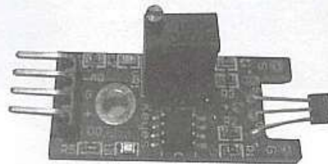


 Linear Hall Magnetic Module

Linear Hall Magnetic Module คือ เซ็นเซอร์ตรวจจับสนามแม่เหล็กเชิงเส้น ซึ่งมีการนำเซ็นเซอร์ชนิดนี้ไปใช้งานร่วมกับเครื่องวัดค่าความเข้มของสนามแม่เหล็ก โดยเราจะนำ Linear Hall Magnetic Module มาต่อใช้งานกับบอร์ด Arduino และทดลองตรวจจับสนามแม่เหล็กที่ปล่อยออกมาจากแม่เหล็กลำโพง

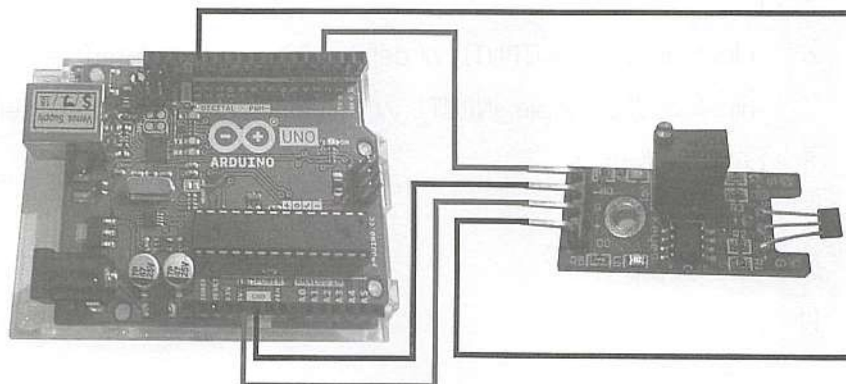
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Linear Hall Magnetic Module
3. Jumper Wire
4. แม่เหล็กลำโพง

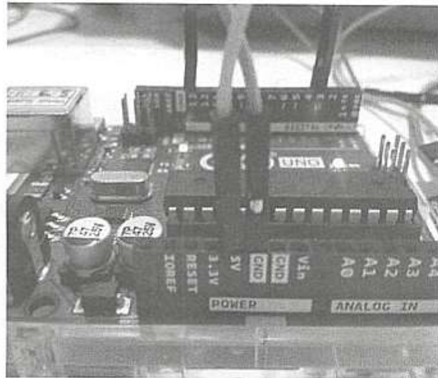


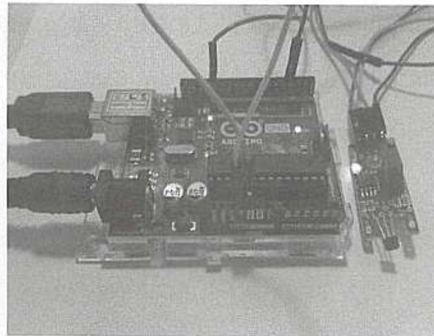
การต่อวงจร

- ขา AO ของเซ็นเซอร์ต่อกับหมุด ~3 บนบอร์ด Arduino
- ขา G ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา DO ของเซ็นเซอร์ต่อกับหมุด D13 บนบอร์ด Arduino



เมื่อเซ็นเซอร์ตรวจจับสนามแม่เหล็กได้ ไฟ LED ในวงจรเซ็นเซอร์จะดับ หากเซ็นเซอร์พ้นสนามแม่เหล็กแล้ว ไฟ LED ในวงจรเซ็นเซอร์จะติด





Sketch Code

```
int Led = 13 ; // define LED Interface
int buttonpin = 3; // define the linear Hall magnetic sensor interface
int val ; // define numeric variables val
void setup ()
{
 pinMode (Led, OUTPUT) ; // define LED as output interface
 pinMode (buttonpin, INPUT) ; // define linear Hall magnetic sensor
output interface
}
void loop ()
{
```

```
val = digitalRead (buttonpin) ; // digital interface will be assigned
a value of 3 to read val
if (val == HIGH) // When the linear Hall sensor detects a magnetic
signal, LED flashes
{
 digitalWrite (Led, HIGH);
}
else
{
 digitalWrite (Led, LOW);
}
}
```

การ Verify (Compile) และการ Upload

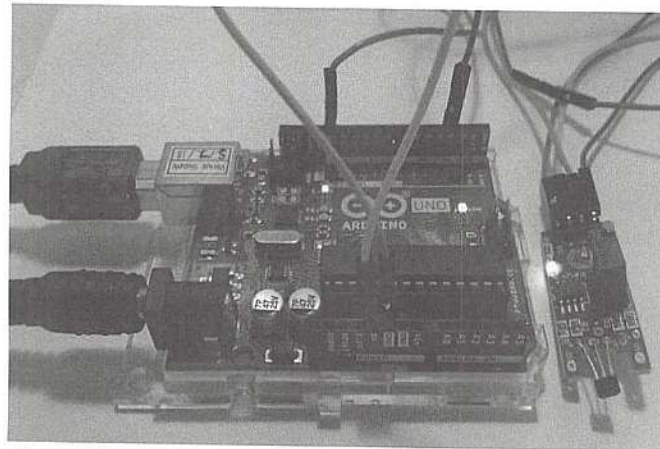


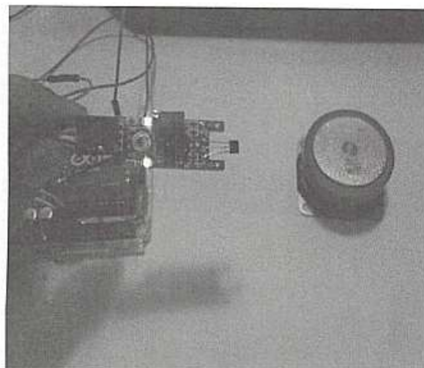
Done compiling.

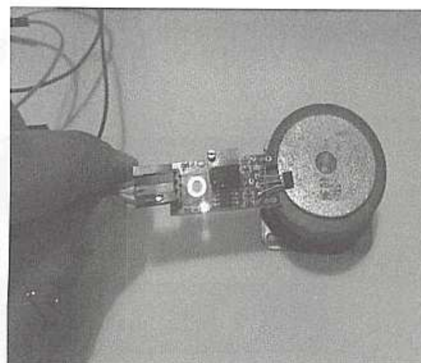


Done uploading.

หลังจาก Upload Sketch Code แล้ว ไฟ LED ในวงจรเซ็นเซอร์จะติด ทั้ง On LED และ S LED แต่หากเซ็นเซอร์สามารถตรวจจับสนามแม่เหล็กได้ ไฟ LED ในวงจรเซ็นเซอร์จะดับ







 Buzzer Module

Buzzer Module คือ โมดูลส่งเสียงสัญญาณเตือน ซึ่งเราสามารถนำไปประยุกต์ใช้ได้ในทุกเซ็นเซอร์โมดูล โดยใช้หลักการที่ว่า เมื่อใดที่เซ็นเซอร์สามารถตรวจจับอะไรได้ตามคุณสมบัติ Buzzer Module จะส่งสัญญาณเตือนทันที เราสามารถนำ Sketch Code ของ Buzzer Module ไปใช้ร่วมกับ Sketch Code ของเซ็นเซอร์โมดูลทุกชุด

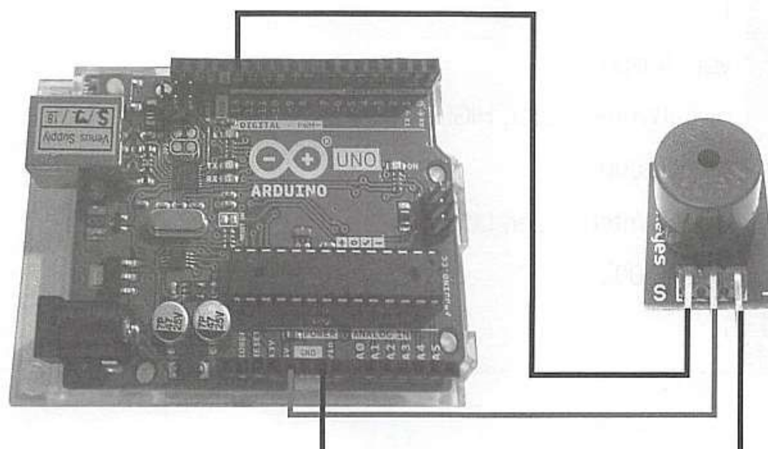
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Buzzer Module
3. Jumper Wire

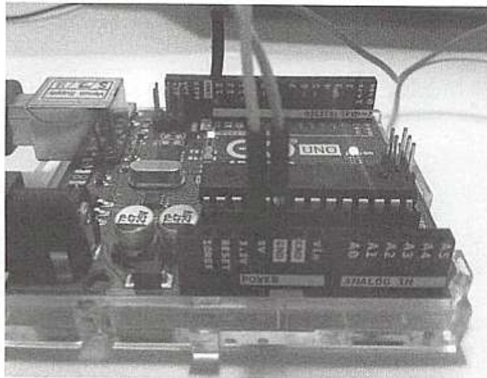


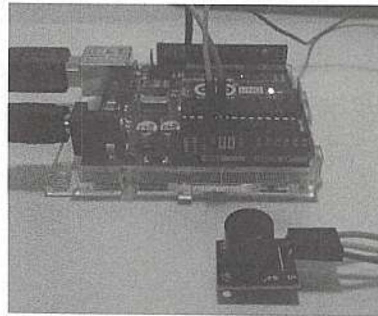
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด D13 บนบอร์ด Arduino
- ขา + ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



Sketch Code จะสั่งให้ Buzzer Module ส่งสัญญาณเตือนตาม delay ที่ได้ตั้งค่าไว้ delay(1000); delay(500);





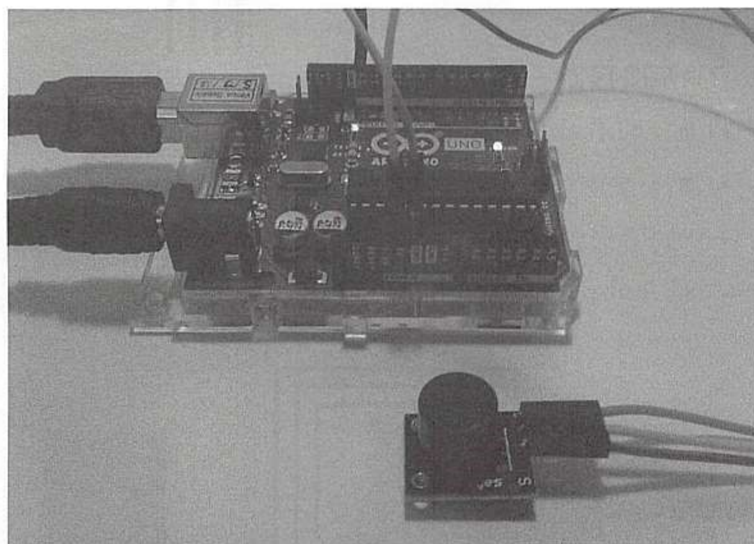
Sketch Code

```
int buzzer = 13;
void setup() {
pinMode(buzzer, OUTPUT);
}
void loop() {
digitalWrite(buzzer, HIGH);
delay(1000);
digitalWrite(buzzer, LOW);
delay(500);
}
```

การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Upload เสร็จเรียบร้อยแล้ว Buzzer Module จะส่งเสียงเตือนตามค่า delay ที่กำหนดไว้ใน Sketch Code ซึ่งเราสามารถนำไปต่อวงจรร่วมกับวงจรโมดูลเซ็นเซอร์อื่นๆ ได้ทันที รวมถึงการเพิ่มเติม Code ของ Buzzer Module ใน Sketch Code อื่นด้วย

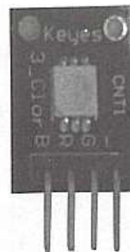


 SMD RGB Module

SMD ย่อมาจาก Surface Mount LED คือ ไฟ LED ที่ยึดติดกับผิวหน้าแผ่นวงจร PCB ซึ่งอุปกรณ์อิเล็กทรอนิกส์โมดูลนี้ จะเป็น LED ชนิดที่สามารถเปล่งแสงได้มากกว่า 1 สี RGB คือ สีแดง (RED) สีเขียว (GREEN) และสีน้ำเงินหรือสีฟ้า (BLUE) โดยเราจะนำมาทำงานร่วมกับบอร์ด Arduino โดยเราจะใช้โปรแกรมบนเว็บไซต์ Arduino Web Editor เพื่อ Upload โปรแกรมไปยังบอร์ด Arduino

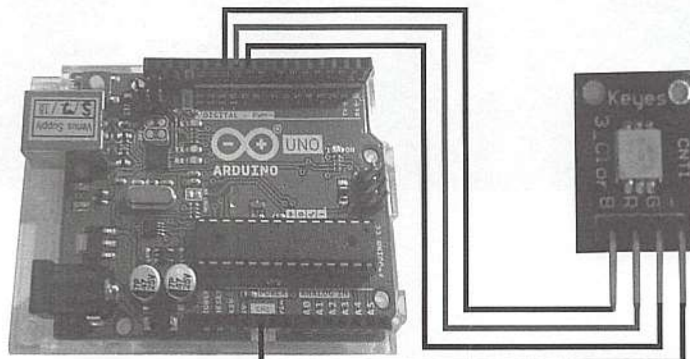
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. SMD RGB Module
3. Jumper Wire

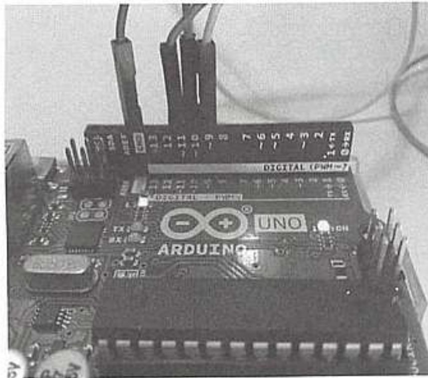


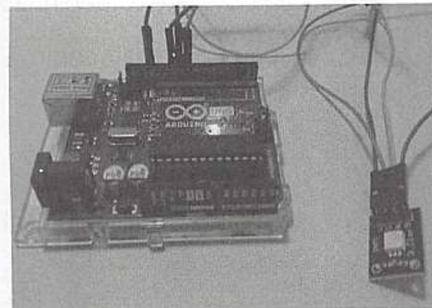
การต่อวงจร

- ขา - ของชุดโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา G ของชุดโมดูลต่อกับหมุด ~9 บนบอร์ด Arduino
- ขา R ของชุดโมดูลต่อกับหมุด ~10 บนบอร์ด Arduino
- ขา B ของชุดโมดูลต่อกับหมุด ~11 บนบอร์ด Arduino



เมื่อ Upload Sketch Code ไปยังบอร์ด Arduino สัญญาณจะถูกส่งออกไปยังหมุด 9, 10 และ 11 ทันที SMD RGB Module จะเปล่งแสงสลับกัน 3 สี ตามเวลาที่กำหนดไว้ใน Sketch Code



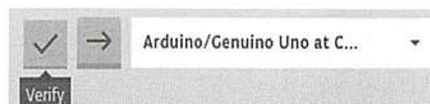


Sketch Code

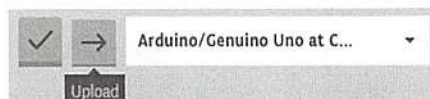
```
int redpin = 11; // select the pin for the red LED
int bluepin = 10; // select the pin for the blue LED
int greenpin = 9; // select the pin for the green LED
int val=0;
void setup () {
 pinMode (redpin, OUTPUT);
 pinMode (bluepin, OUTPUT);
 pinMode (greenpin, OUTPUT);
 Serial.begin (9600);
}
```

```
void loop () {
 for (val=255; val>0; val--)
 {
  analogWrite (redpin, val);
  analogWrite (bluepin, 255-val);
  analogWrite (greenpin, 128-val);
  delay (1);
 }
 for (val = 0; val <255; val++)
 {
  analogWrite (redpin, val);
  analogWrite (bluepin, 255-val);
  analogWrite (greenpin, 128-val);
  delay (1);
 }
 Serial.println (val, DEC);
}
```

การ Verify (Compile) และการ Upload

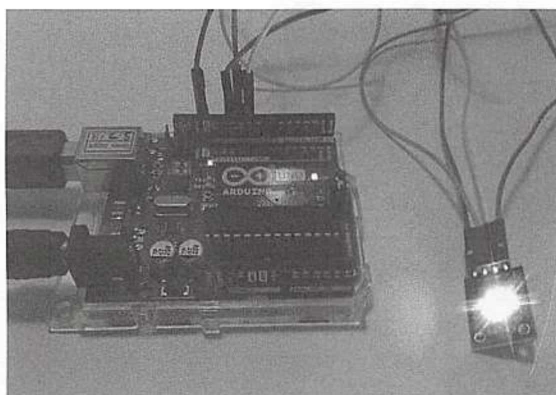


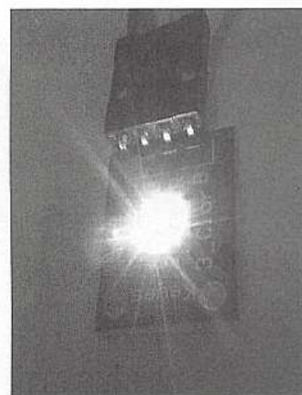
Success: Done verifying



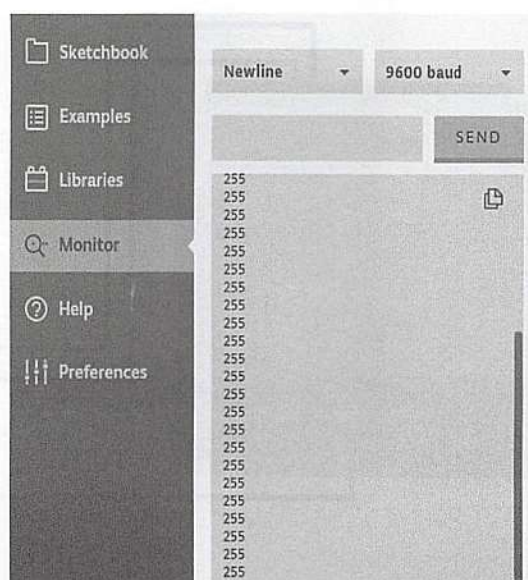
Success: Done uploading

หลังจากที่ระบบ Upload โปรแกรม Sketch Code ไปยังบอร์ด Arduino "Success" แล้ว หากเราต่อวงจร SMD RGB Module ถูกต้อง ไฟ LED จะเปล่งแสงสลับกัน 3 สีทันที โดยเราได้มีการกำหนดระยะเวลาใน Sketch Code ให้ติดและดับที่ ติด 255 มิลลิวินาที ดับ 128 มิลลิวินาที และวนซ้ำต่อไปเรื่อยๆ





โดยหากเราคลิกเลือกที่เมนู Monitor ด้านซ้ายมือ ระบบจะแสดงค่าเวลาการติดของไฟ LED เป็นมิลลิวินาที ตามที่เราได้กำหนดไว้ใน Sketch Code

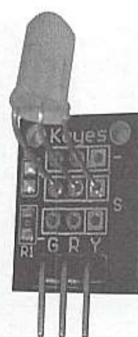


 2-Color LED Module

2-Color LED Module คือ ชุดไฟ LED ที่สามารถเปล่งแสงได้ 2 สี คือ สีแดงและสีเขียว โดยเราจะเขียนโปรแกรมชุดคำสั่งให้ไฟ LED ติดสลับสีกัน ตามเวลาที่เราได้กำหนดไว้ใน Sketch Code

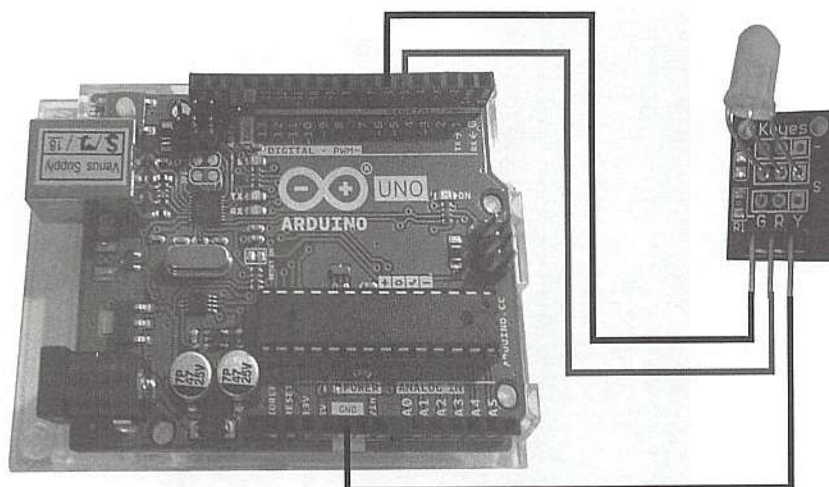
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. 2-Color LED Module
3. Jumper Wire
4. Breadboard

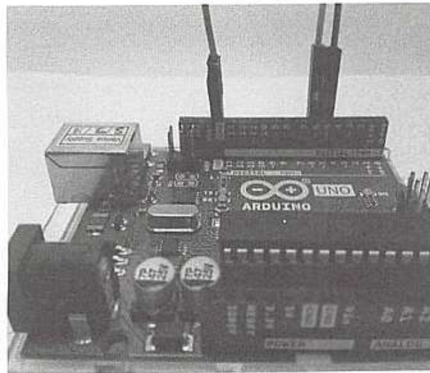


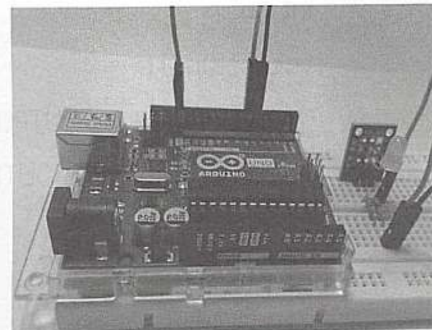
การต่อวงจร

- ขา R ของโมดูลต่อกับหมุด ~5 บนบอร์ด Arduino
- ขา G ของโมดูลต่อกับหมุด ~6 บนบอร์ด Arduino
- ขา Y ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino



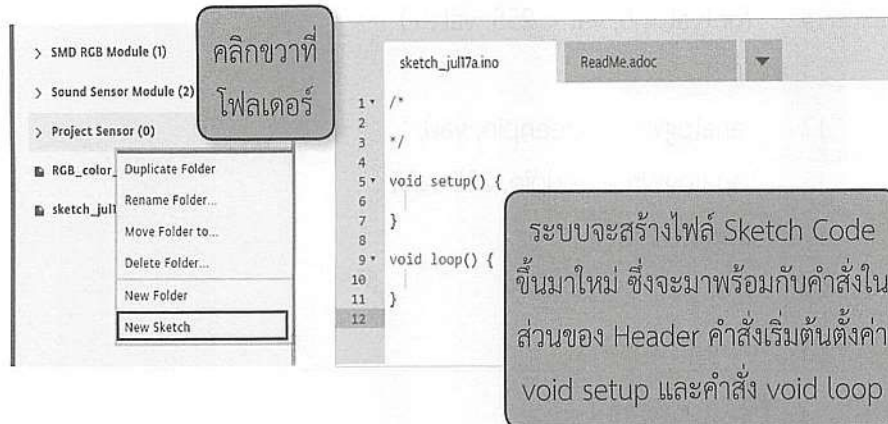
เมื่อ Upload โปรแกรม Sketch Code บอร์ด Arduino จะส่งสัญญาณไปที่หมุดที่กำหนดไว้ใน Sketch Code ทันที 2-Color LED Module จะเปล่งแสงสลับกันสองสี ตามเวลาที่ได้หน่วงเวลาไว้ใน Sketch Code





Sketch Code

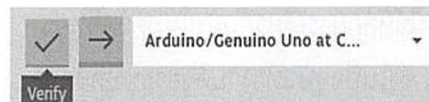
ให้คลิกขวาโฟลเดอร์ที่สร้างไว้เพื่อเก็บ Sketch Code จากนั้นเลือก New Sketch เพื่อสร้าง Sketch Code ใหม่



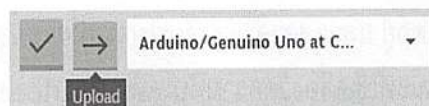
จากนั้นก็เริ่มทำการพิมพ์ชุดคำสั่งได้เลย

```
int redpin = 5; // select the pin for the red LED
int greenpin = 6; // select the pin for the green LED
int val;
void setup () {
  pinMode (redpin, OUTPUT);
  pinMode (greenpin, OUTPUT);
}
void loop () {
  for (val = 255; val > 0; val--)
   {
   analogWrite (greenpin, val);
   analogWrite (redpin, 255-val);
   delay (15);
  }
  for (val = 0; val < 255; val++)
   {
   analogWrite (greenpin, val);
   analogWrite (redpin, 255-val);
   delay (15);
  }
}
```

การ Verify (Compile) และการ Upload

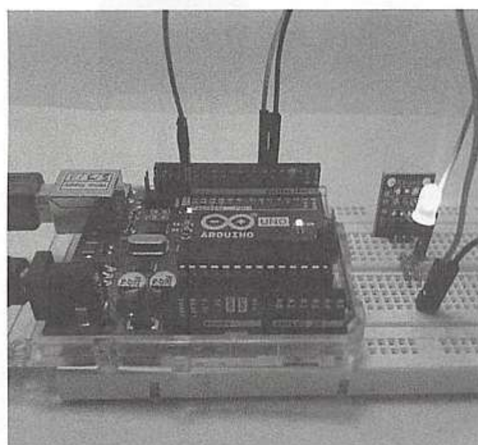


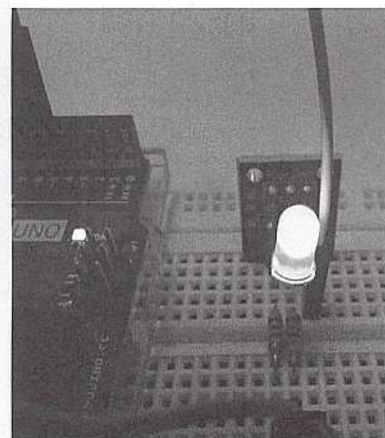
Success: Done verifying



Success: Done uploading

หลังจากที่ระบบทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว บอร์ดจะส่งสัญญาณออกจากหมุด 5 และหมุด 6 ทำให้ไฟ LED สีเขียวติด 255 มิลลิวินาที และสีแดงติด 255 มิลลิวินาที แล้วหน่วงเวลาไว้ 15 มิลลิวินาที ทำงานเช่นนี้วนซ้ำต่อไปเรื่อยๆ



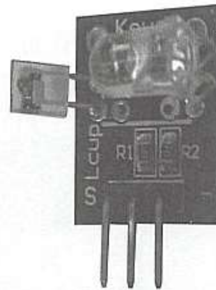


 Heartbeat Sensor Module

Heartbeat Sensor Module คือ โมดูลเซ็นเซอร์ที่สามารถตรวจวัดสัญญาณชีพ "ชีพจร" ของมนุษย์ได้ ซึ่งใช้หลักการเดียวกันกับโหมดวัดชีพจร หรือ Pulse ในเครื่องวัดความดันโลหิต ซึ่งในปัจจุบันทำงานด้วยระบบดิจิตอล โดยทำงานผ่านหลักการปล่อยแสงอินฟราเรดผ่านนิ้วมือของมนุษย์ ซึ่งด้านใต้ของนิ้วจะมีเซ็นเซอร์ตรวจจับแสงอินฟราเรดอยู่ และแสดงค่าแสงอินฟราเรดที่ผ่านนิ้วมือนั้น เพื่อตรวจวัดค่าชีพจร ทั้งนี้ในขณะที่ใช้เครื่องวัดความดันโลหิต Heartbeat Sensor Module จะตรวจวัดชีพจรพร้อมกันไปด้วย โดยเราจะจะนำโมดูลนี้มาทำงานร่วมกับบอร์ด Arduino เพื่อแสดงค่าที่เซ็นเซอร์สามารถตรวจวัดได้

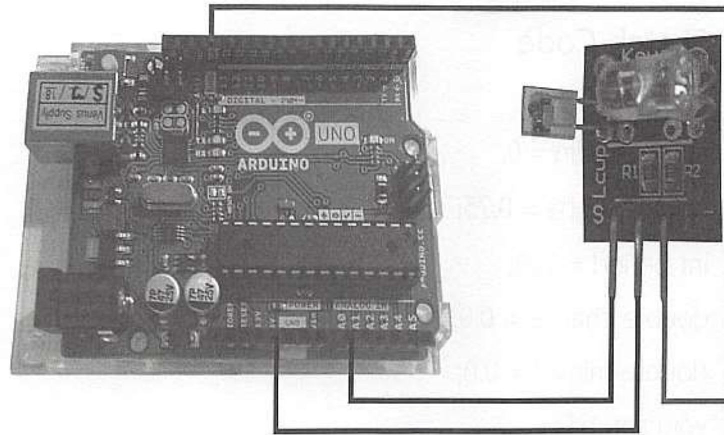
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Heartbeat Sensor Module
3. Jumper Wire
4. นิ้วมือมนุษย์

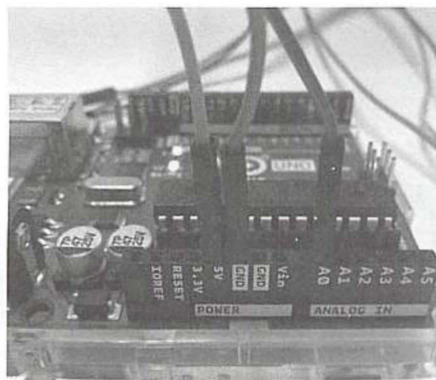


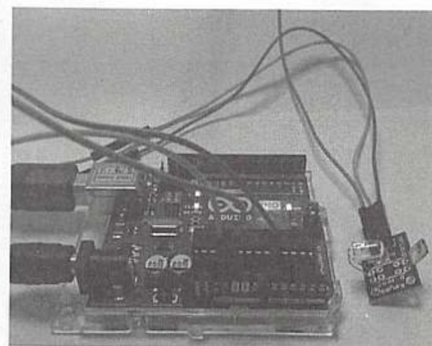
การต่อวงจร

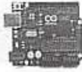
- ขา S ของเซ็นเซอร์ต่อกับหมุด AO บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อสอดนิ้วมือเข้าไปในเซ็นเซอร์ โมดูลจะตรวจวัดชีพจรที่ปลายนิ้วมือของเรา และแสดงค่าเป็นอะนาล็อก โดยเราจะสามารถดูค่าตัวเลขได้ที่ Monitor บนเว็บไซต์ Arduino Web Editor และสามารถดู Serial Plotter ซึ่งจะแสดงเป็นเส้นกราฟได้ที่โปรแกรม Arduino IDE นั่นแสดงว่าเราจะต้อง Upload Sketch Code จากโปรแกรม Arduino IDE ด้วย เพื่อดูเส้นกราฟการทดลองตรวจวัดสัญญาณชีพจรของเรา







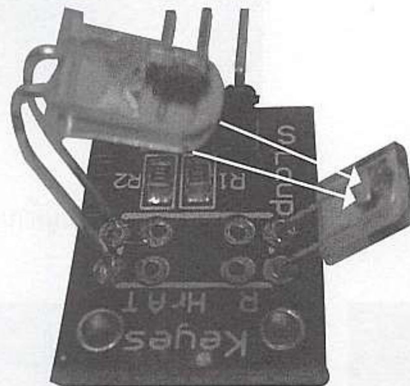
Sketch Code

```
int sensorPin = 0;
double alpha = 0.75;
int period = 100;
double change = 0.0;
double minval = 0.0;
void setup () {
Serial.begin (9600);
}
void loop () {
static double oldValue = 0;
static double oldChange = 0;
int rawValue = analogRead (sensorPin);
double value = alpha * oldValue + (1 - alpha) * rawValue;
Serial.print (rawValue);
Serial.print (“,”);
Serial.println (value);
oldValue = value;
delay (period);
}
```

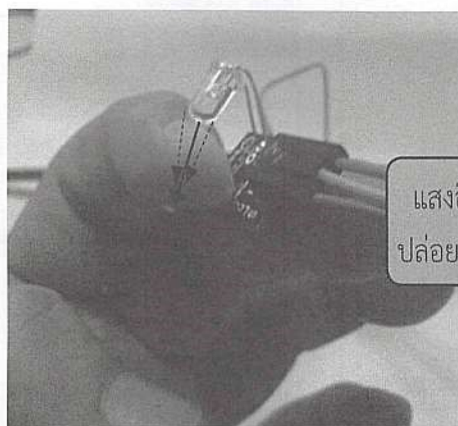
การ Verify (Compile) และการ Upload

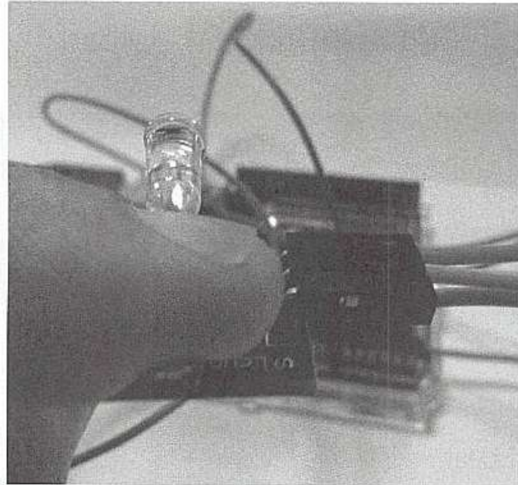


ก่อนที่จะทำการทดสอบการทำงานของ Heartbeat Sensor Module เราจะต้องทำการปรับหลอดอินฟราเรดและเซ็นเซอร์ให้อยู่ในตำแหน่งที่ตรงกัน

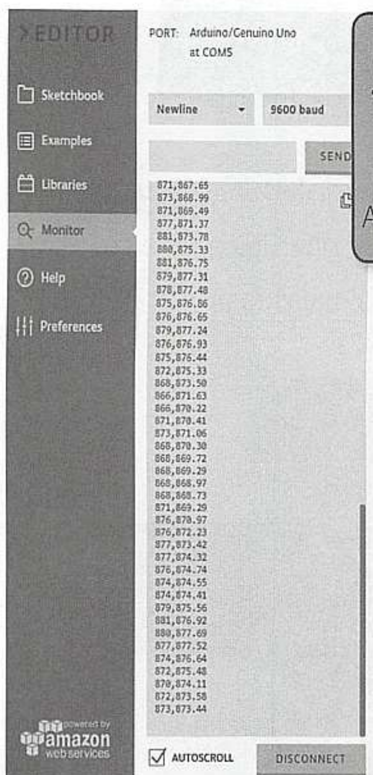


เพื่อให้เวลาที่เราเอานิ้วมือสอดเข้าไป แสงอินฟราเรดจะปล่อยผ่านนิ้วมือของเรา และตรงไปยังเซ็นเซอร์เพื่อตรวจวัดค่า หลังจากที่ Upload Sketch Code ไปยังบอร์ด Arduino ลองสอดนิ้วมือเข้าไปในเซ็นเซอร์ได้เลย

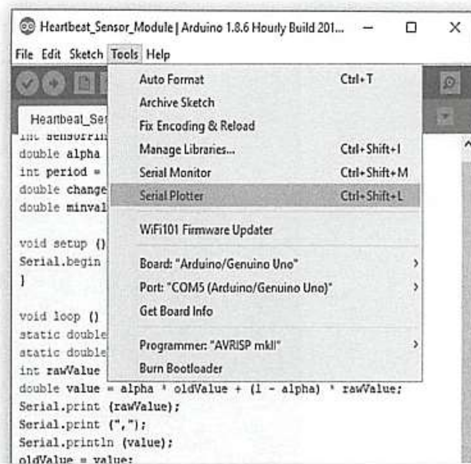


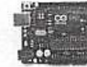


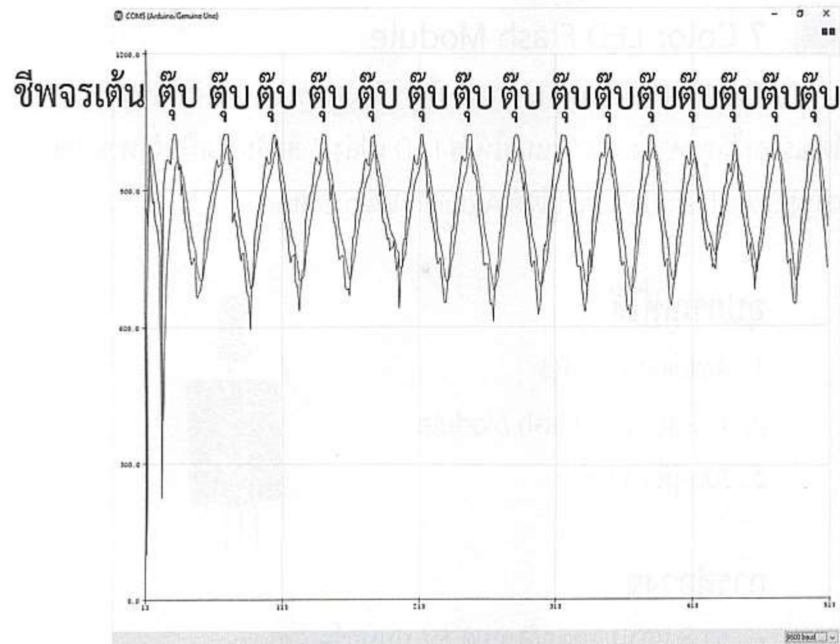
จากนั้นดูค่าใน Monitor เมนูด้านซ้ายบนเว็บไซต์ Arduino Web Editor

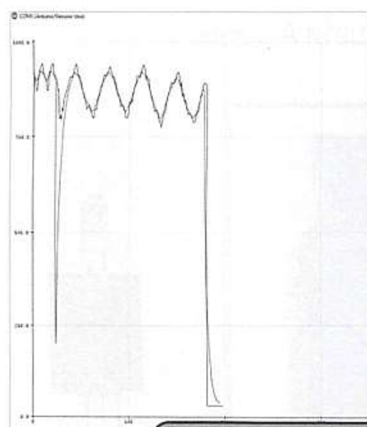


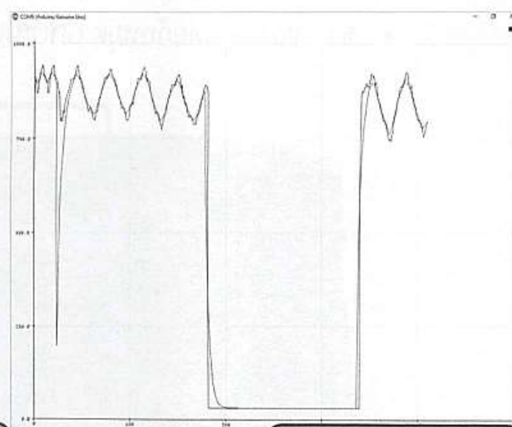
หรือดูเส้นกราฟค่าชีพจร ใน Serial Plotter ในโปรแกรม Arduino IDE โดยเมื่อ Upload Sketch Code จากโปรแกรม Arduino IDE ไปยังบอร์ด Arduino แล้ว ให้ไปที่ Tools เลือกที่ Serial Plotter











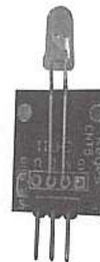
ถอดนิ้วมือออกจากเซ็นเซอร์ Death

สอดนิ้วมือเข้าเซ็นเซอร์ตามเดิม Revive

 7 Color LED Flash Module

7 Color LED Flash Module คือ ชุดโมดูลไฟ LED 7 สี ในหลอดเดียว โดยเราจะใช้ชุดคำสั่ง เพื่อกำหนดให้ไฟ LED เปล่ง 7 สีสลับกันเป็นจังหวะ โดยสร้าง Sketch Code บนเว็บไซต์ Arduino Web Editor

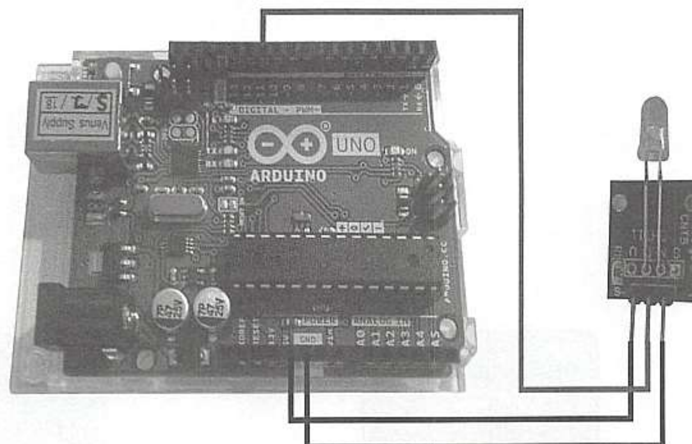
อุปกรณ์ที่ใช้



1. Arduino Uno R3
2. 7 Color LED Flash Module
3. Jumper Wire

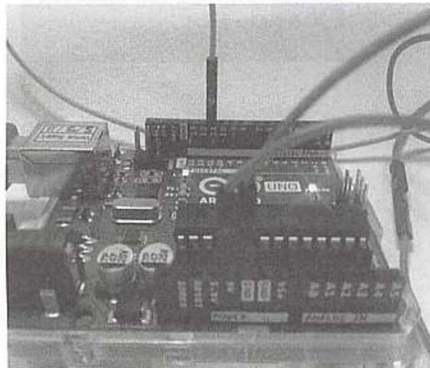
การต่อวงจร

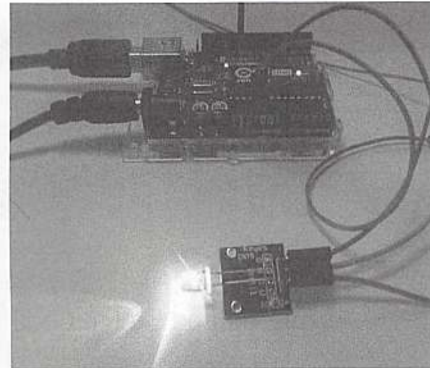
- ขา S ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขากลางของโมดูลต่อกับหมุด ~11 บนบอร์ด Arduino
- ขา - ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino



อย่าพึ่งสงสัยว่า ทำไมขา S Data ถึงไปต่อกับหมุด 5V แล้วคิดว่าตัวอย่างต่อผิด แล้วสลับเอาขากลางไปต่อ 5V แทน บอร์ดจะพังเอานะจ๊ะ (สำหรับโมดูล Keyes เท่านั้นนะ)

บอร์ด Arduino จะส่งสัญญาณออกจากหมุด 11 เพื่อป้อนโปรแกรมคำสั่งให้ 7 Color LED Flash Module ทำงานตามโปรแกรม

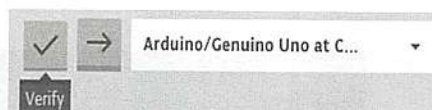




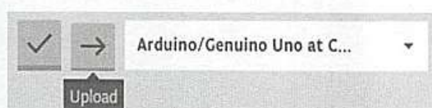
Sketch Code

```
void setup () {
 // Initialize the digital pin as an output.
 // Pin 11 has an LED connected on most Arduino boards:
 pinMode (11, OUTPUT);
}
void loop () {
 digitalWrite (11, HIGH); // set the LED on
 delay (2000); // wait for a second
 digitalWrite (11, LOW); // set the LED off
 delay (2000); // wait for a second
}
```

การ Verify (Compile) และการ Upload

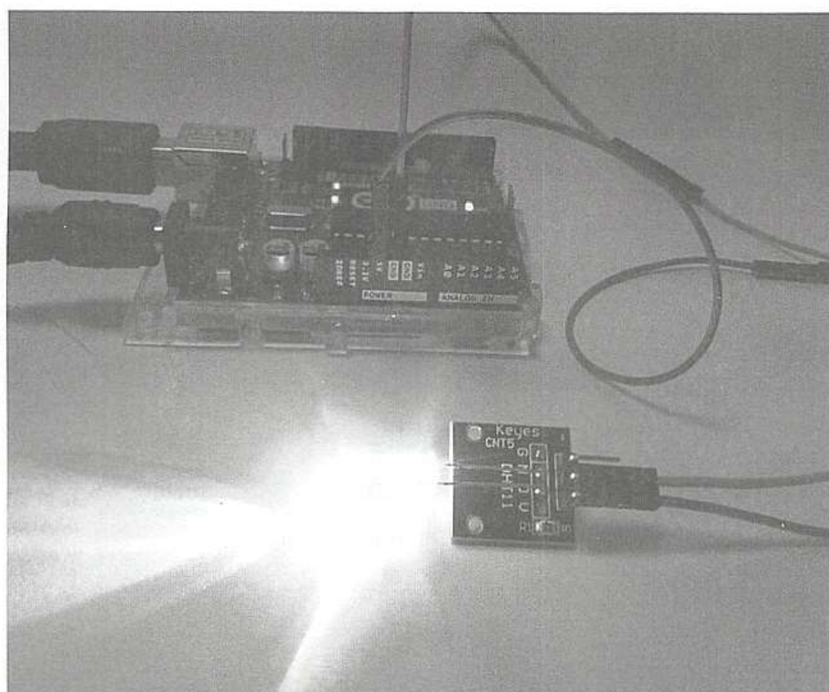


Success: Done verifying



Success: Done uploading

เมื่อ Upload Sketch Code ไปยังบอร์ด Arduino ชุดโปรแกรมคำสั่ง จะกำหนดให้ไฟ LED แต่ละสีติดในเวลา 2,000 มิลลิวินาที และดับในเวลา 2,000 มิลลิวินาที จากนั้นจะกะพริบทั้ง 7 สี 3 รอบ แล้วกลับไปวนซ้ำใหม่ตั้งแต่เริ่มต้น



 Switch Button Module

Switch Button Module คือ วงจรปุ่มสวิตช์ง่ายๆ เช่นเดียวกับปุ่มสวิตช์บนบอร์ด Arduino (ปุ่มข้างพอร์ต USB) โดยการทำงาน เมื่อกดปุ่มจะดับ ปล่อยปุ่มจะติด ทั้งนี้สวิตช์โมดูลนี้จะใช้งานร่วมกับแพลตฟอร์มอิเล็กทรอนิกส์ยอดนิยมอย่าง Arduino, Raspberry Pi และ Esp8266 เพื่อเป็นการ Reset ระบบการทำงาน ซึ่งเราจะนำมาทดลองรับคำสั่งจากบอร์ด Arduino

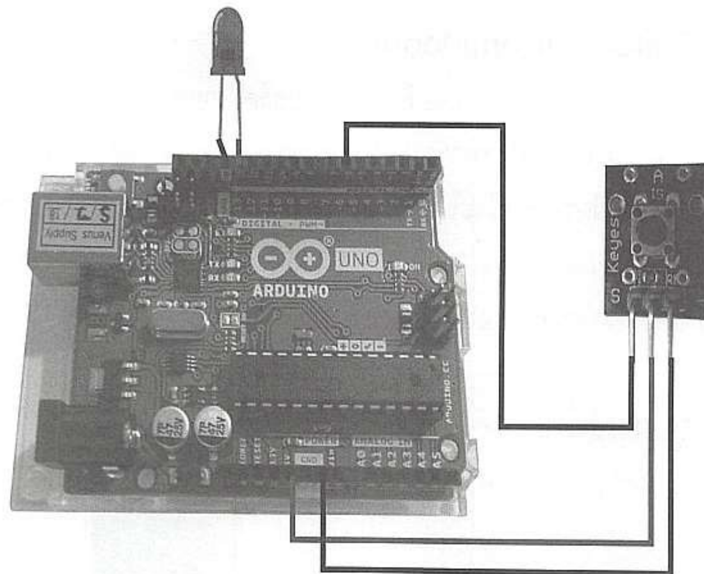
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Switch Button Module
3. Jumper Wire
4. LED
5. Breadboard

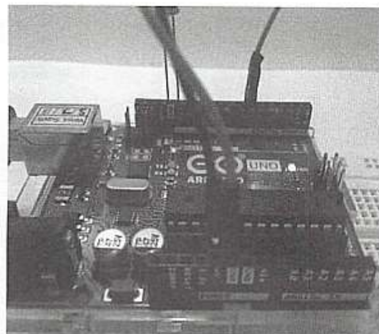


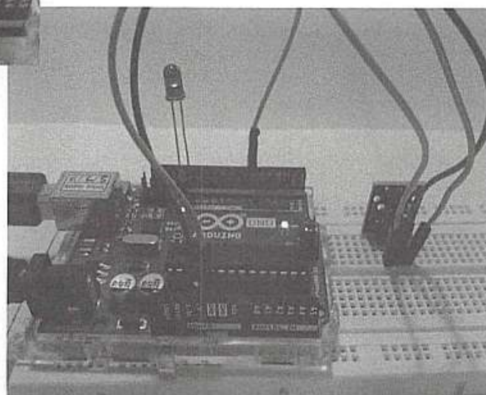
การต่อวงจร

- ขา S ของโมดูลต่อกับหมุด ~6 บนบอร์ด Arduino
- ขากลางของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด 13 บนบอร์ด Arduino
- ขา LED - ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อชุดโปรแกรมคำสั่ง Upload ไปยังบอร์ด Arduino เมื่อใดที่กดปุ่มบนโมดูล ไฟ LED จะดับ เมื่อปล่อยปุ่มไฟ LED จะติดตามเดิม

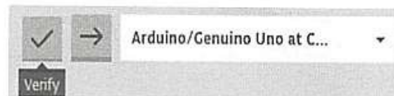




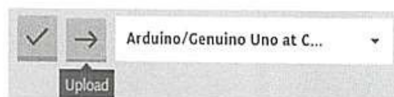
Sketch Code

```
int Led = 13 ;// Declaration of the LED-output pin
int Key = 3; // Declaration of the sensor input pin
int val; // Temporary variable
void setup ()
{
 pinMode (Led, OUTPUT) ; // Initialization output pin
 pinMode (Key, INPUT_PULLUP) ; // Initialization Key pin with internal
 pull-up resistor
}
void loop ()
{
 val = digitalRead (Key) ; // The current signal at the key will be read
 if (val == HIGH) // If a signal was detected, the LED will light up.
 {
  digitalWrite (Led, HIGH);
 }
 else
 {
  digitalWrite (Led, LOW);
 }
}
```

การ Verify (Compile) และการ Upload

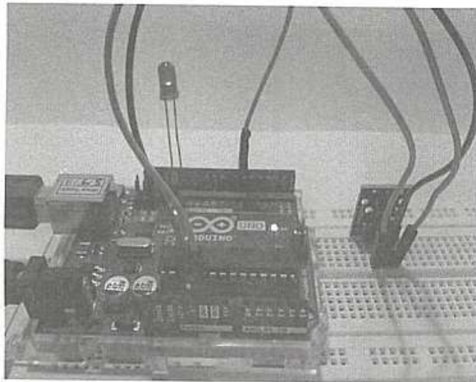


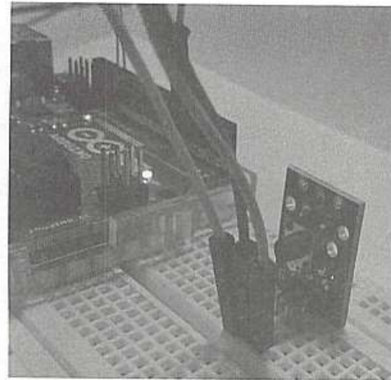
Success: Done verifying

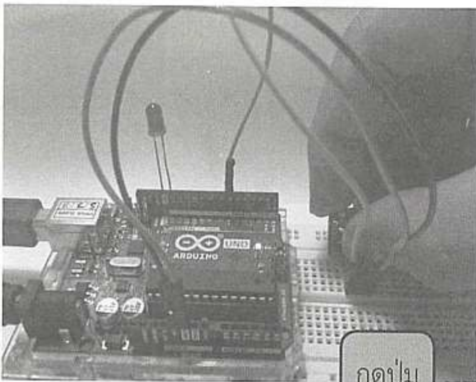


Success: Done uploading

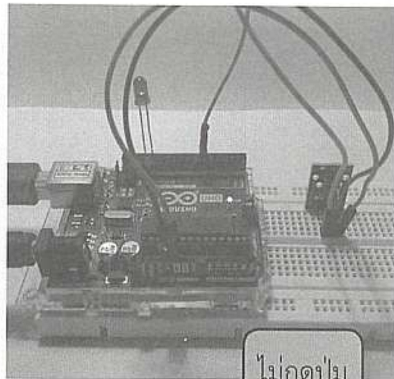
หลังจากที่ทำการ Upload โปรแกรมชุดคำสั่งไปยังบอร์ด Arduino แล้ว ไฟ LED ที่รับสัญญาณจากหมุด 13 ของบอร์ด Arduino จะติดทันที เช่นนั้นเมื่อใดที่เรากดปุ่ม ไฟ LED จะดับ และเมื่อใดที่เราปล่อยปุ่ม ไฟ LED จะติดตามปกติ







กดปุ่ม



ไม่กดปุ่ม

 Shock Sensor Module

Shock Sensor Module คือ เซ็นเซอร์ตรวจจับการสั่นสะเทือนที่เกิดขึ้นกับโมดูลเซ็นเซอร์ โดยเมื่อใดที่หลอดเซ็นเซอร์มีการสั่นสะเทือน จะส่งสัญญาณไปยังบอร์ด Arduino ทันที จากนั้นบอร์ดจะทำการประมวลผลและแสดงเป็นคำเตือน Shock Alarm หรือ No Alarm

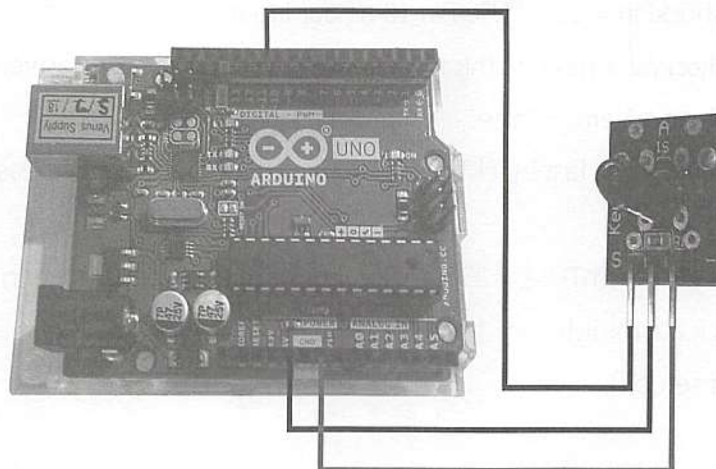
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Shock Sensor Module
3. Jumper Wire

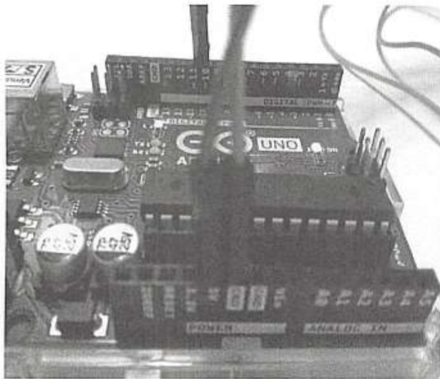


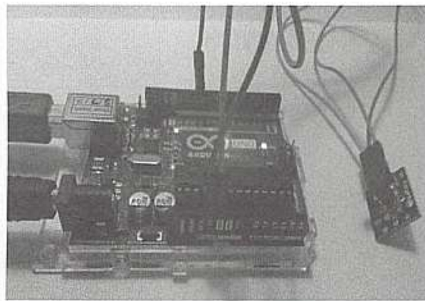
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~11 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



โดยในตัวเซ็นเซอร์จะมีสปริง 2 ขั้ว หากเกิดการสั่นสะเทือนขึ้นกับเซ็นเซอร์ สปริงจะแกว่งไปถูกขั้ว เมื่อเซ็นเซอร์ตรวจวัดความสั่นสะเทือนได้ จะส่งสัญญาณไปที่บอร์ด Arduino ผ่านขา S ไปหมุด 11 บอร์ดจะทำการประมวลผลและแสดงออกมาเป็น Shock Alarm หรือ No Alarm บน Serial Monitor



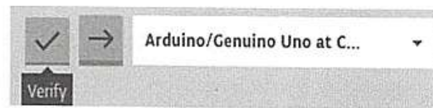


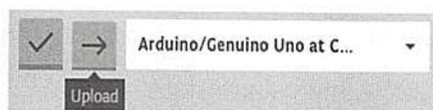
Sketch Code

```
int shockPin = 11; // Use Pin 10 as our Input
int shockVal = HIGH; // This is where we record our shock measurement
boolean bAlarm = false;
unsigned long lastShockTime; // Record the time that we meas-
ured a shock
int shockAlarmTime = 250; // Number of milli seconds to keep the
shock alarm high
void setup ()
{
```

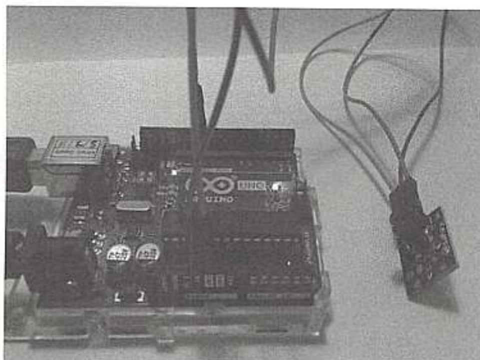
```
  Serial.begin(9600);
  pinMode (shockPin, INPUT) ; // input from the KY-002
}
void loop ()
{
  shockVal = digitalRead (shockPin) ; // read the value from our sensor
  if (shockVal == LOW) // If we're in an alarm state
  {
   lastShockTime = millis(); // record the time of the shock
   // The following is so you don't scroll on the output screen
   if (!bAlarm){
    Serial.println("Shock Alarm");
    bAlarm = true;
   }
  }
  else
  {
   if( (millis()-lastShockTime) > shockAlarmTime && bAlarm){
    Serial.println("no alarm");
    bAlarm = false;
   }
  }
}
```

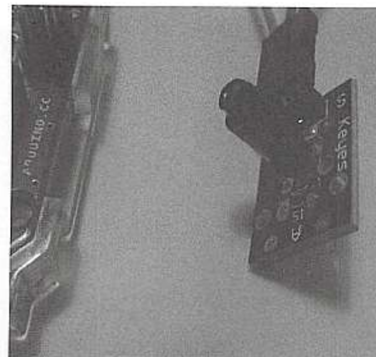
การ Verify (Compile) และการ Upload





เมื่อทำให้เกิดการสั่นสะเทือนกับเซ็นเซอร์ โมดูลเซ็นเซอร์จะส่งสัญญาณไปที่บอร์ด Arduino ซึ่งเราจะสามารถดูการเตือนได้ที่เมนู Monitor ด้านซ้ายบนเว็บไซต์ Arduino Web Editor







 Mini Reed Switch

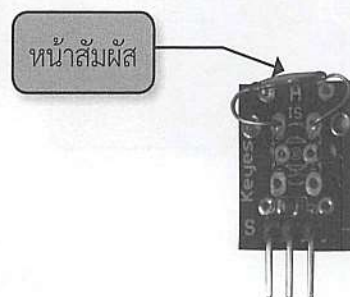
Mini Reed Switch คือ สวิตช์ที่ทำงานได้ในสนามแม่เหล็ก โดยปกติหน้าสัมผัสของโมดูลจะแยกออกจากกัน แต่เมื่อใดที่สวิตช์อยู่ในอำนาจของสนามแม่เหล็ก ไม่ว่าจะเป็นอำนาจแม่ที่เกิดจากประจุไฟฟ้า หรือเป็นโหละที่มีคุณสมบัติของแม่เหล็ก หน้าสัมผัสจะถูกดูดให้ติดกัน สวิตช์จะทำการต่อวงจรทันที

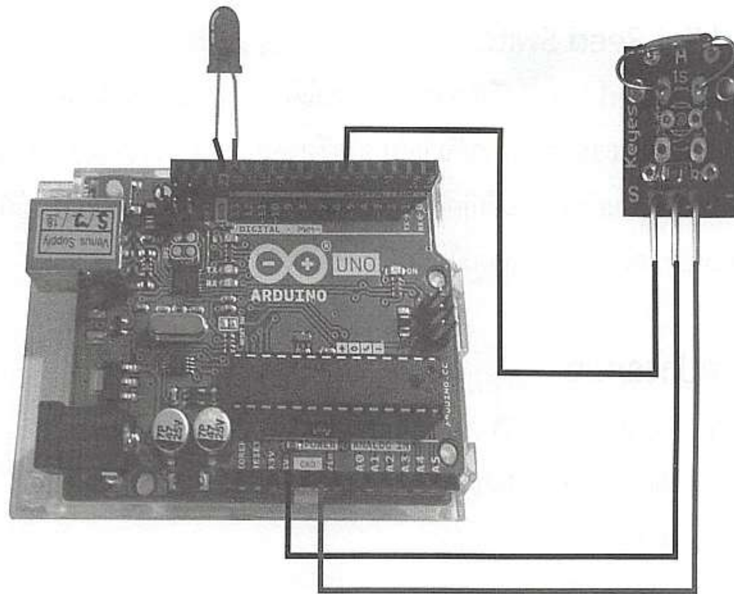
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Mini Reed Switch
3. LED
4. Jumper Wire
5. Breadboard
6. แม่เหล็ก

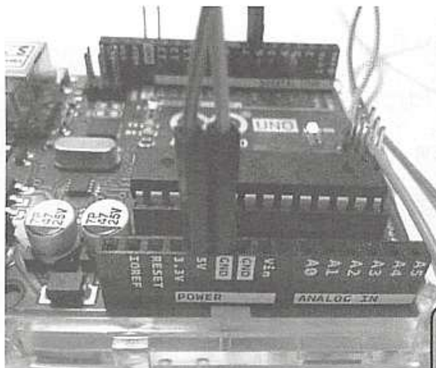
การต่อวงจร

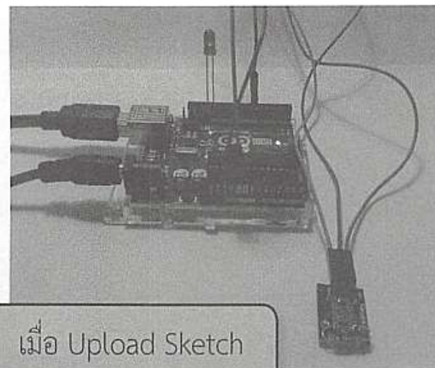
- ขา S ของเซ็นเซอร์ต่อกับหมุด ~6 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด 13 บนบอร์ด Arduino
- ขา LED - ต่อกับหมุด GND บนบอร์ด Arduino





เมื่อเซ็นเซอร์อยู่ในสนามแม่เหล็ก เซ็นเซอร์จะส่งสัญญาณไปที่บอร์ด Arduino ผ่านขา S ไปยังหมุด 6 จึงทำให้บอร์ด Arduino ตัดสัญญาณหมุด 13 ซึ่งส่งผลให้ไฟ LED ดับ



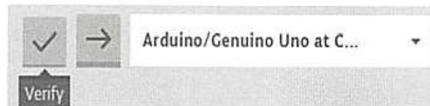


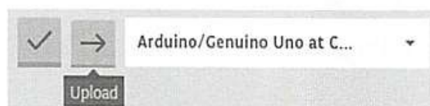
เมื่อ Upload Sketch Code ไฟ LED จะติดทันที

Sketch Code

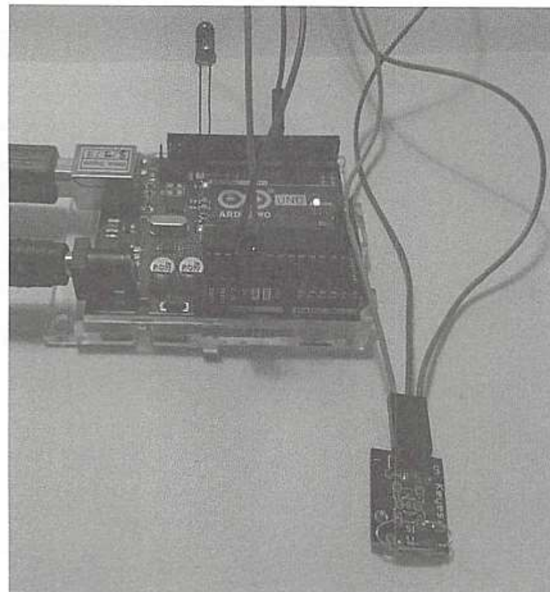
```
// Example code for sensor KY021
int Led = 13 ;// define LED Interface
int buttonpin = 6; // define the Reed sensor interfaces
int val ;// define numeric variables val
void setup ()
{
 pinMode (Led, OUTPUT) ;// define LED as output interface
 pinMode (buttonpin, INPUT) ;// output interface as defined
Reed sensor
}
void loop ()
{
 val = digitalRead (buttonpin) ;// digital interface will be as-
signed a value of 3 to read val
 if (val == HIGH) // When the Reed sensor detects a signal,
LED flashes
 {
  digitalWrite (Led, HIGH);
 }
 else
 {
  digitalWrite (Led, LOW);
 }
}
```

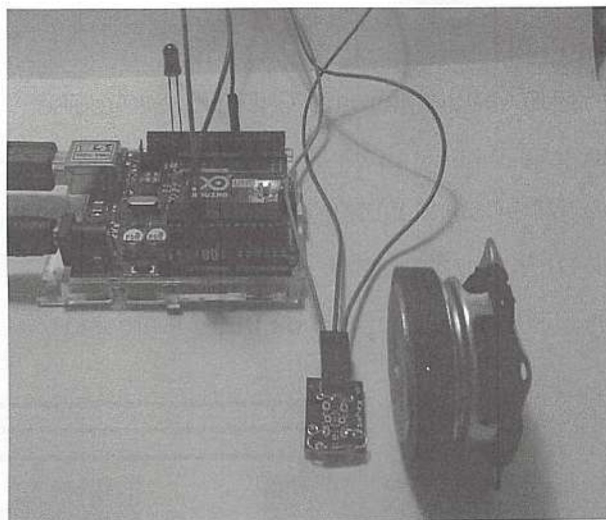
การ Verify (Compile) และการ Upload





เมื่อระบบทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว ไฟ LED ที่เชื่อมต่อกับหมุด 13 จะเปล่งแสงขึ้นทันที แต่เมื่อใดที่นำแม่เหล็กเข้าไปใกล้ตัวเซ็นเซอร์ หน้าสัมผัสของ Reed Switch จะติดกัน และโมดูลจะส่งสัญญาณไปที่บอร์ด Arduino โดยบอร์ดจะตัดสัญญาณหมุด 13 ส่งผลให้ไฟ LED ดับ "Reed Switch ติด ไฟ LED บนบอร์ดดับ"





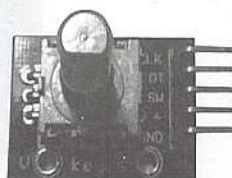


Rotary Encoder Module

Rotary Encoder Module คือ เซ็นเซอร์ที่มีคุณสมบัติในการตรวจจับการเคลื่อนไหวของมอเตอร์ หรือเซอร์โว โดย Rotary Encoder จะเป็นเครื่องเข้ารหัสโรตารี่ ซึ่งมีรูปแบบสัญญาณเอาต์พุตที่ออกมาเป็นลักษณะพัลส์ที่เป็นรูปสี่เหลี่ยม ซึ่งสัญญาณที่ถูกส่งออกมา จะมีความสัมพันธ์กับระยะการเคลื่อนที่ ตำแหน่ง ระยะห่าง ความเร็ว และความเร่ง รวมถึงทิศทางการหมุน เช่น ตามเข็มหรือทวนเข็มนาฬิกา โดย Rotary Encoder Module นี้ จะเป็นเครื่องเข้ารหัสโรตารี่ ซึ่งเป็นส่วนประกอบอิเล็กทรอนิกส์ที่สามารถตรวจจับการเคลื่อนไหวและตำแหน่งเมื่อหมุนได้ ทั้งนี้เราจะนำ Rotary Encoder Module มาทำงานภายใต้ Sketch Code ของบอร์ด Arduino

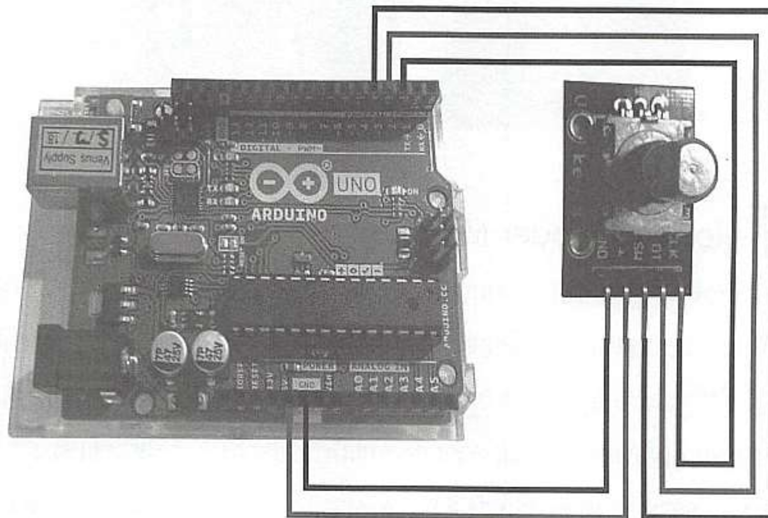
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Rotary Encoder Module
3. Jumper Wire

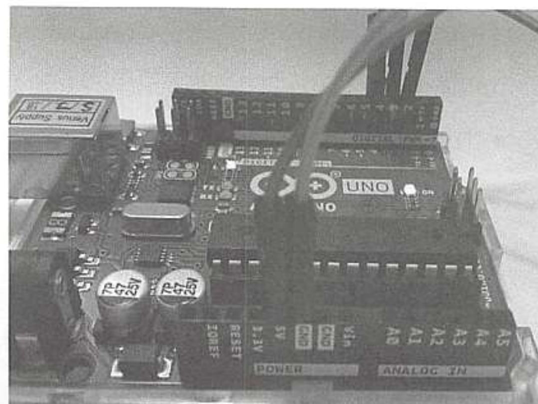


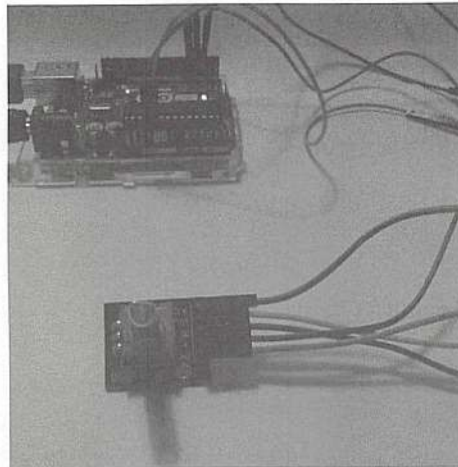
การต่อวงจร

- ขา GND ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา SW ของโมดูลต่อกับหมุด D4 บนบอร์ด Arduino
- ขา DT ของโมดูลต่อกับหมุด ~3 บนบอร์ด Arduino
- ขา CLK ของโมดูลต่อกับหมุด D2 บนบอร์ด Arduino



โดย Rotary Encoder Module จะส่งสัญญาณไปที่บอร์ด Arduino เพื่อแสดงค่า Test บน Serial Monitor



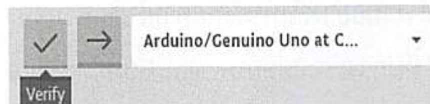


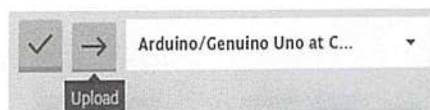
Sketch Code

```
#define encoder0PinA 2
#define encoder0PinB 3
#define encoder0Btn 4
int encoder0Pos = 0;
void setup() {
 Serial.begin(9600);
 pinMode(encoder0PinA, INPUT_PULLUP);
 pinMode(encoder0PinB, INPUT_PULLUP);
 pinMode(encoder0Btn, INPUT_PULLUP);
 attachInterrupt(0, doEncoder, CHANGE);
}
int valRotary,lastValRotary;
void loop() {
 int btn = digitalRead(encoder0Btn);
```

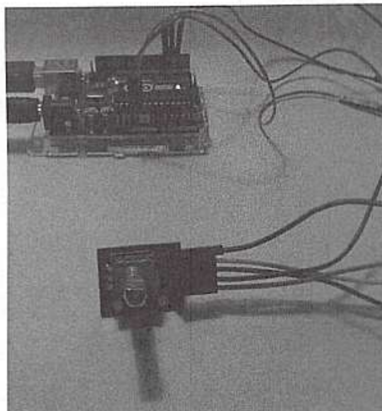
```
 Serial.print(btn);
 Serial.print(" ");
 Serial.print(valRotary);
 if(valRotary>lastValRotary)
 {
      Serial.print(" CW");
 }
 if(valRotary) {
      Serial.print(" CCW");
 }
 lastValRotary = valRotary;
 Serial.println(" ");
 delay(250);
}
void doEncoder()
{
 if (digitalRead(encoder0PinA) == digitalRead(encoder0PinB))
 {
      encoder0Pos++;
 }
 else
 {
      encoder0Pos--;
 }
 valRotary = encoder0Pos/2.5;
}
```

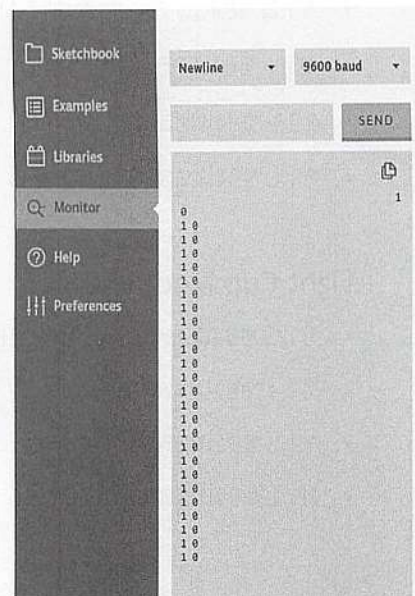
การ Verify (Compile) และการ Upload





เมื่อระบบ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว เซ็นเซอร์จะส่งสัญญาณข้อมูลไปที่บอร์ด Arduino ซึ่งเราจะสามารถดูค่าที่ Serial Monitor ที่เมนู Monitor ด้านซ้ายมือ



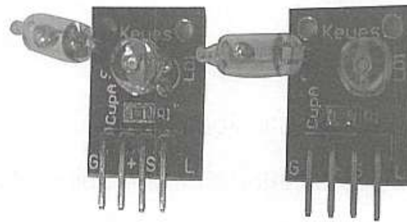


 Light Cup Module

Light Cup Module คือ ชุดไฟ LED ที่ควบคุมการเปิด-ปิด ด้วยปรอทในกระเปาะแล้ว โดยจะมีโมดูล 2 ชุด ซึ่งทำงานอย่างสัมพันธ์กัน ตามโปรแกรมชุดคำสั่งที่ Upload ไปยังบอร์ด Arduino ที่มีการเชื่อมต่อกับ Light Cup Module

อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Light Cup Module
3. Jumper Wire



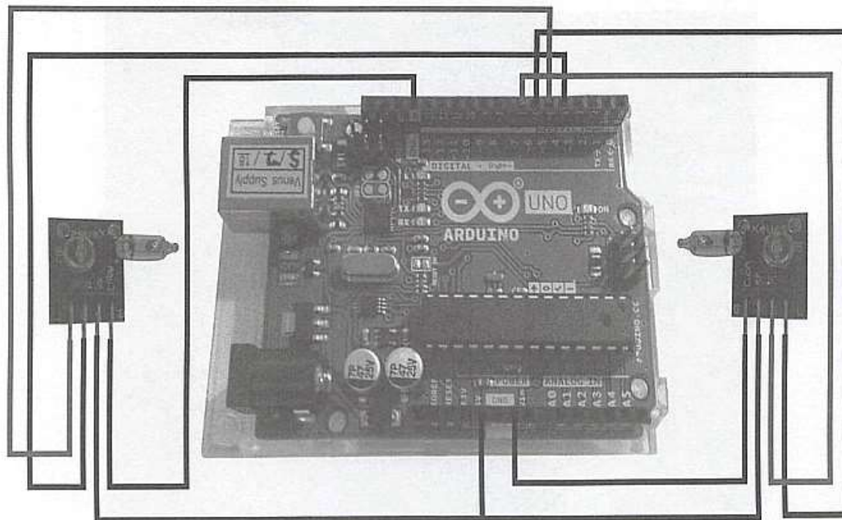
การต่อวงจร

Light Cup Module 1

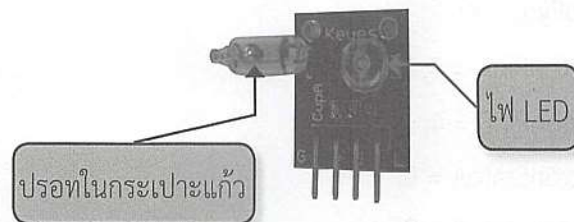
- ขา G ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา S ของโมดูลต่อกับหมุด D4 บนบอร์ด Arduino
- ขา L ของโมดูลต่อกับหมุด ~5 บนบอร์ด Arduino

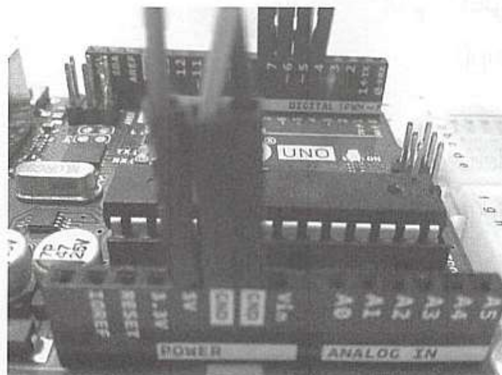
Light Cup Module 2

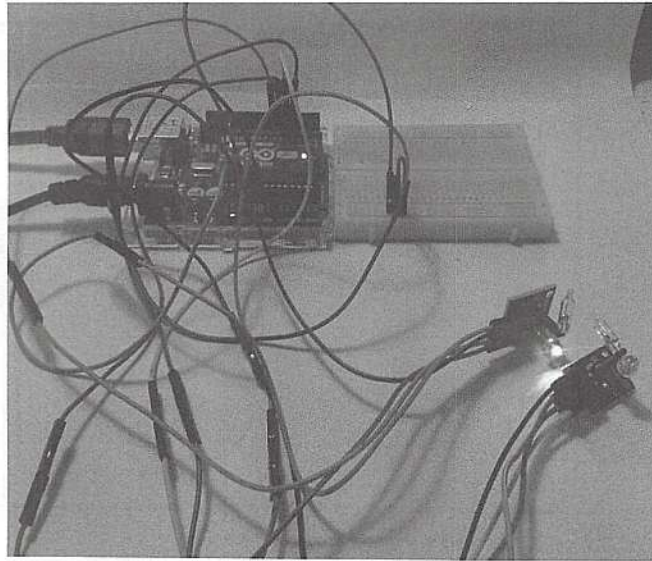
- ขา G ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา + ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา S ของโมดูลต่อกับหมุด D7 บนบอร์ด Arduino
- ขา L ของโมดูลต่อกับหมุด ~6 บนบอร์ด Arduino



เมื่อทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว ไฟ LED ชุดที่ 1 จะติด จากนั้นเมื่อเอียงให้ปรอทไหลลงมาแตะที่ขั้วอีกด้านหนึ่ง ไฟ LED ชุดที่ 2 จะติด







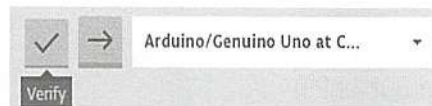
Sketch Code

```
int LedPinA = 5;
int LedPinB = 6;
int ButtonPinA = 7;
int ButtonPinB = 4;
int buttonStateA = 0;
int buttonStateB = 0;
int brightness  = 0;
 void setup()
{
  pinMode(LedPinA, OUTPUT);
  pinMode(LedPinB, OUTPUT);
  pinMode(ButtonPinA, INPUT);
  pinMode(ButtonPinB, INPUT);
```

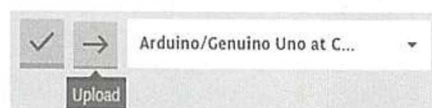
```
}
void loop()
{
 buttonStateA = digitalRead(ButtonPinA);
 if (buttonStateA == HIGH && brightness != 255)
 {
  brightness ++;
 }
  buttonStateB = digitalRead(ButtonPinB);
  if (buttonStateB == HIGH && brightness != 0)
 {
  brightness --;
 }
   analogWrite(LedPinA, brightness); // A will turn off gradually

   analogWrite(LedPinB, 255 - brightness); // B will turn on
gradually
   delay(25);
}
```

การ Verify (Compile) และการ Upload

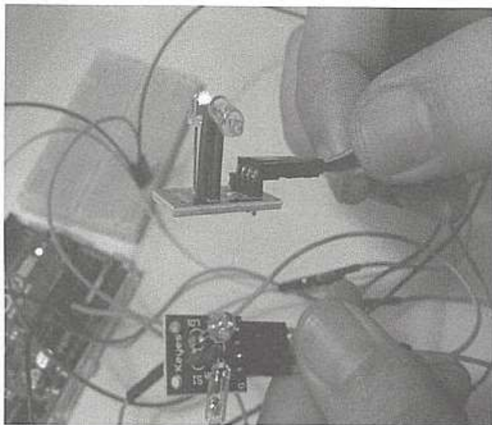


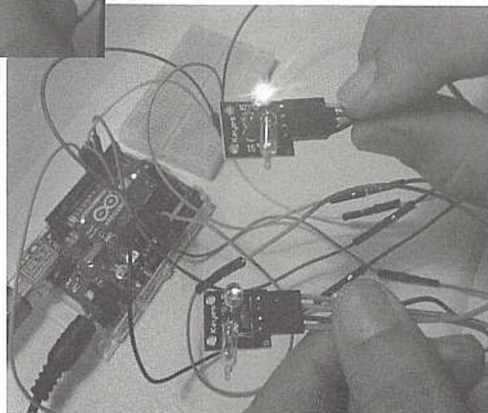
Success: Done verifying

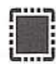


Success: Done uploading

หลังจาก Upload Sketch Code แล้ว ไฟ LED ทั้งสองชุดจะถูกควบคุมด้วยปรอทในกระเปาะแก้ว โดยการเปล่งแสงของไฟ LED สองชุด เปลี่ยนไปในขณะที่เอียงลง คล้ายกับการเทไฟจาก LED หนึ่งไปยัง LED อีกดวงหนึ่ง

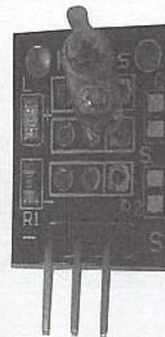




 Tilt Switch Sensor Module

Tilt Switch Sensor Module คือ สวิตช์เซ็นเซอร์ที่ควบคุมด้วยปรอทในกระเปาะแก้ว โดยนำไปใช้เกี่ยวกับการทรงตัวของสมาร์ตคาร์ หรือหุ่นยนต์ ให้สามารถเคลื่อนที่ได้เป็นเส้นตรง ทั้งนี้ Tilt Switch Sensor Module จะทำงานโดยอาศัยหลักการเคลื่อนที่ของปรอท ซึ่งไหลอยู่ระหว่าง 2 ขั้วในกระเปาะแก้ว

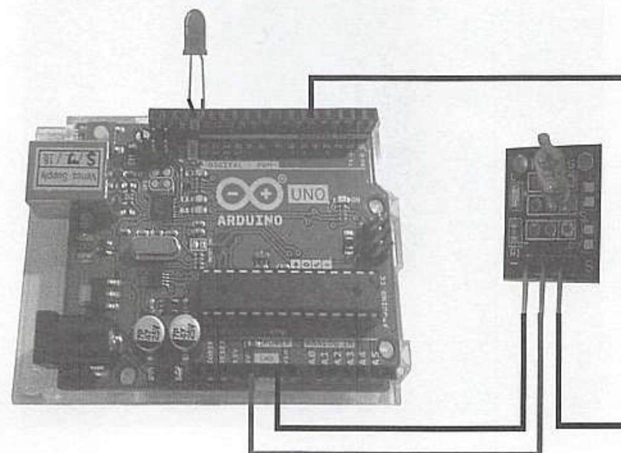
อุปกรณ์ที่ใช้



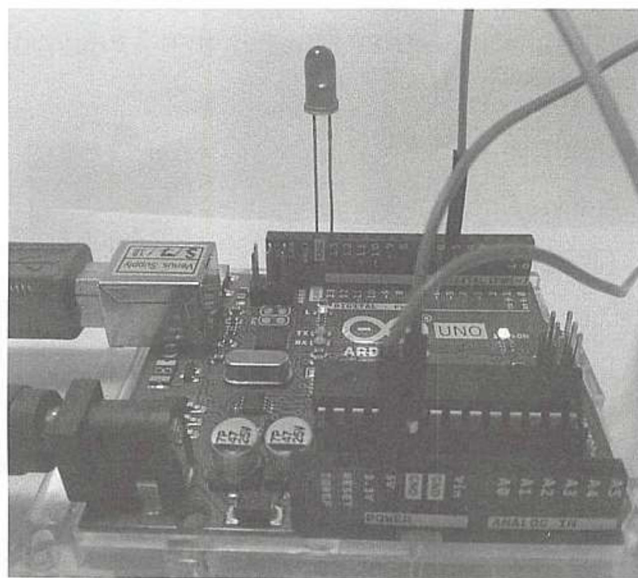
1. Arduino Uno R3
2. Tilt Switch Sensor Module
3. LED
4. Jumper Wire

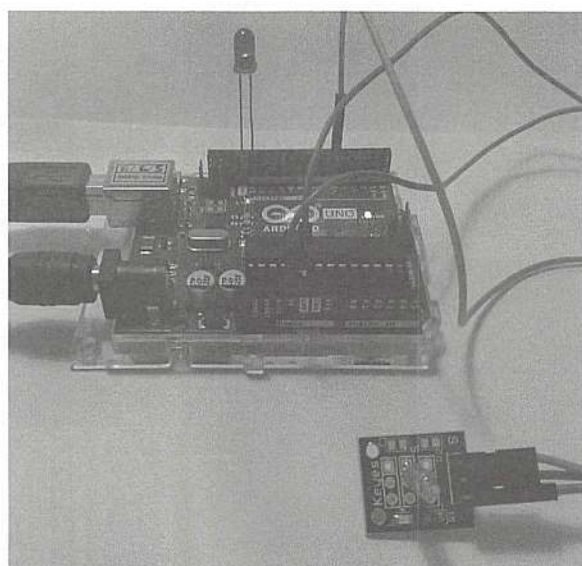
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~5 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ของเซ็นเซอร์ต่อกับหมุด D13 บนบอร์ด Arduino
- ขา LED - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์อยู่ในตำแหน่งที่ปรอทไม่ไหลไปสัมผัสกับขั้วทั้งสองด้าน เซ็นเซอร์จะส่งสัญญาณไปที่บอร์ด Arduino จากนั้นบอร์ดจะตัดสัญญาณหมุด 13 ส่งผลให้ไฟ LED ดับ

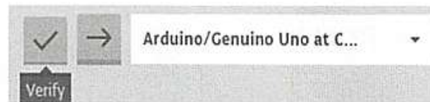




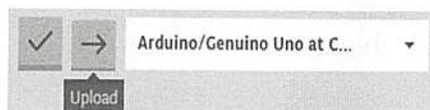
Sketch Code

```
//KY017 Mercury open optical module
int Led = 13 ;// define LED Interface
int buttonpin = 5; // define the mercury tilt switch sensor interface
int val ;// define numeric variables val
void setup ()
{
pinMode (Led, OUTPUT) ;// define LED as output interface
pinMode (buttonpin, INPUT) ;// define the mercury tilt switch
sensor output interface
}
void loop ()
{
val = digitalRead (buttonpin) ;// read the values assigned to
the digital interface 3 val
if (val == HIGH) // When the mercury tilt switch sensor detects
a signal, LED flashes
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
}
```

การ Verify (Compile) และการ Upload

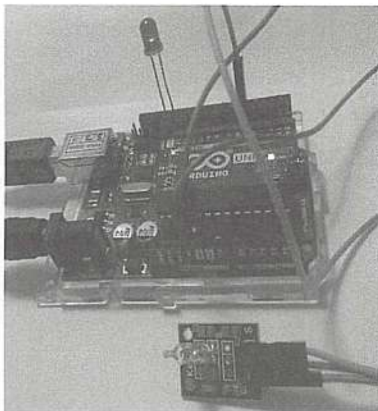


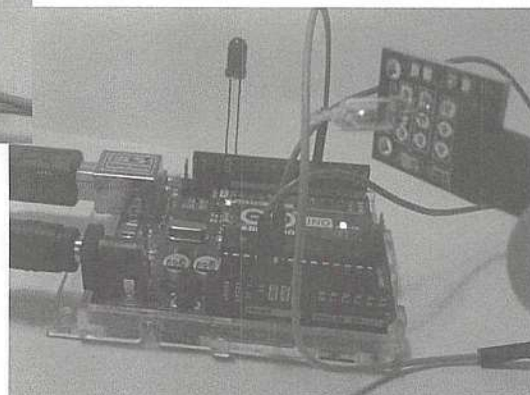
Success: Done verifying



Success: Done uploading

เมื่อทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว ไฟ LED ที่รับสัญญาณจากหมุด 13 จะติดทันที เมื่อประคองเซ็นเซอร์ให้อยู่ในตำแหน่งที่ปรอทในกระเปาะแก้ว ไม่สัมผัสกับขั้วใด ไฟ LED ที่รับสัญญาณจากหมุด 13 จะดับ





 Ball Switch Tilt Sensor

Ball Switch Tilt Sensor คือ สวิตช์เซ็นเซอร์ที่มีหลักการทำงานคล้ายกับ Tilt Switch เพียงแต่ใช้บอลแทนปรอทเป็นหน้าสัมผัส ซึ่งจะอยู่ภายในตัวเซ็นเซอร์และหุ้มไว้ด้วยฉนวนอย่างดี

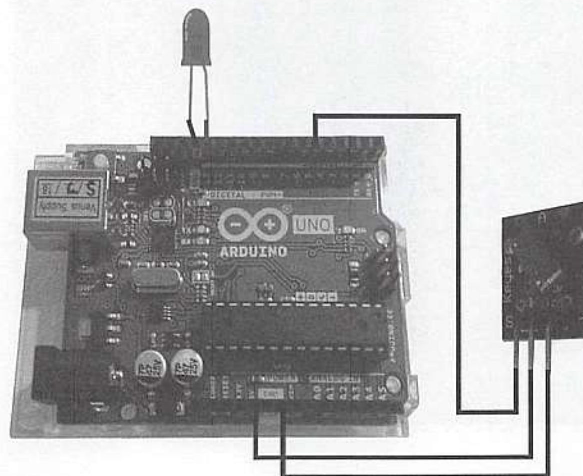
อุปกรณ์ที่ใช้



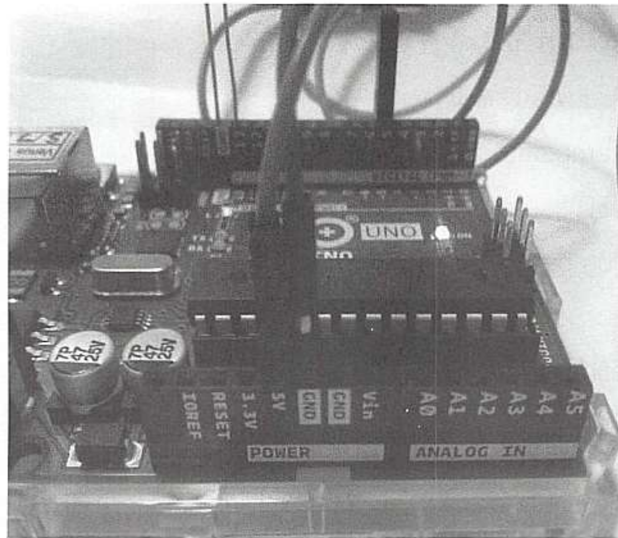
1. Arduino Uno R3
2. Temperature Sensor Module
3. LED
4. Jumper Wire

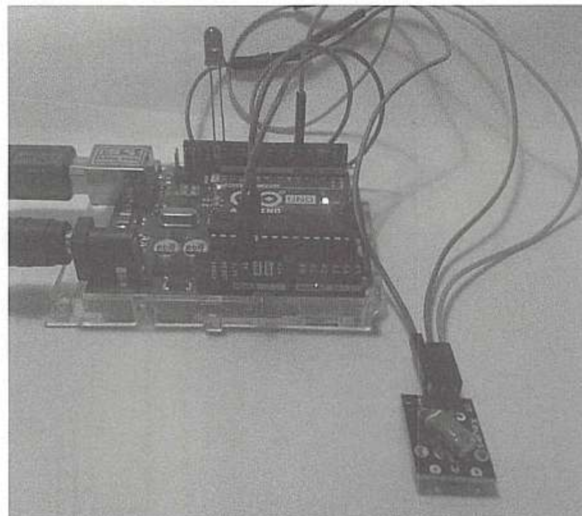
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~5 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ของเซ็นเซอร์ต่อกับหมุด D13 บนบอร์ด Arduino
- ขา LED - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อเซ็นเซอร์อยู่ในตำแหน่งที่ลูกบอลไม่ไหลไปสัมผัสกับขั้วอีกด้าน เซ็นเซอร์จะส่งสัญญาณไปที่บอร์ด Arduino จากนั้นบอร์ดจะตัดสัญญาณหมุด 13 ส่งผลให้ไฟ LED ดับ นั่นแสดงว่า การทดสอบจะเอียงเซ็นเซอร์เพื่อให้ลูกบอลไหลจากขั้วหนึ่งไปยังอีกขั้วหนึ่ง

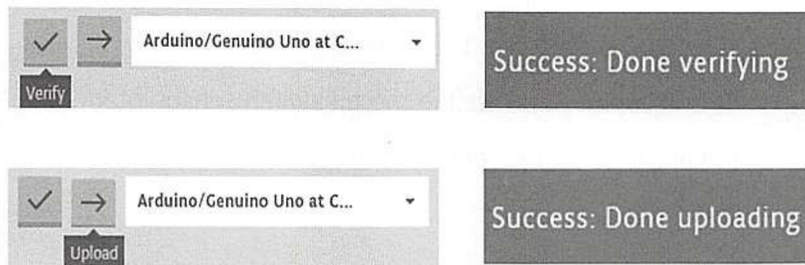




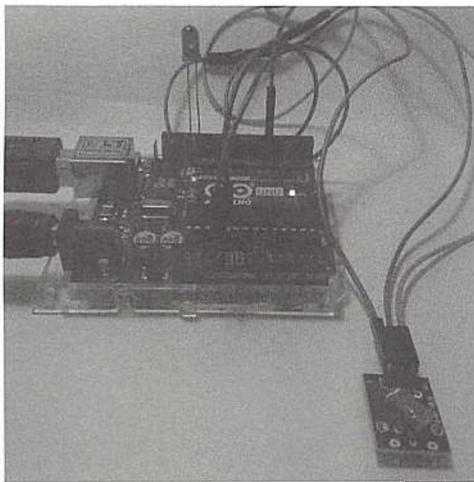
Sketch Code

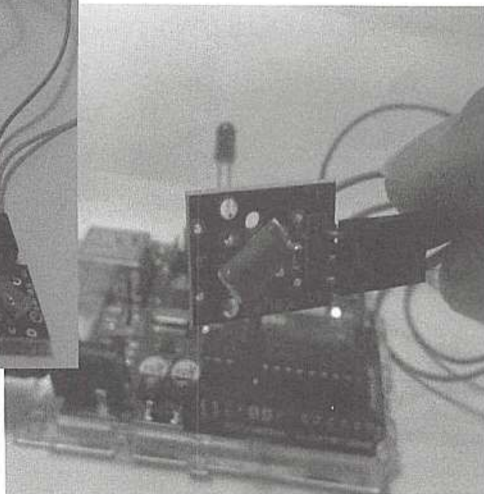
```
int Led = 13 ;// define LED Interface
int buttonpin = 5; // define the tilt switch sensor interfaces
int val ;// define numeric variables val
void setup ()
{
 pinMode (Led, OUTPUT) ;// define LED as output interface
 pinMode (buttonpin, INPUT) ;//define the output interface tilt
switch sensor
}
void loop ()
{
 val = digitalRead (buttonpin) ;// digital interface will be as-
signed a value of 3 to read val
 if (val == HIGH) //When the tilt sensor detects a signal when
the switch, LED flashes
 {
  digitalWrite (Led, HIGH);
 }
 else
 {
  digitalWrite (Led, LOW);
 }
}
```

การ Verify (Compile) และการ Upload



เมื่อทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว ไฟ LED ที่รับสัญญาณจากหมุด 13 จะติดทันที เมื่อเอียงเซ็นเซอร์ให้อยู่ในตำแหน่งที่ลูกบอลในกระบอกเซ็นเซอร์สัมผัสกับอีกขั้วหนึ่ง ไฟ LED ที่รับสัญญาณจากหมุด 13 จะดับ





 Photoresistor Sensor Module

Photoresistor Sensor Module คือ โมดูลเซ็นเซอร์วัดแสง โดยมีส่วนประกอบหลัก คือ LDR หรือตัวต้านทานไวแสง ซึ่งเราหลายคนอาจเคยผ่านหูผ่านตามาบ้างแล้ว กับวงจรเซ็นเซอร์ควบคุมแสง "แสงหมดหลอดไฟติด แสงมาหลอดไฟดับ" แต่ Photoresistor Sensor Module วงจรนี้ ถูกสร้างให้ส่งสัญญาณอะนาล็อก เพื่อแสดงค่าเป็นตัวเลข โดยปริมาณของแสงที่ LDR ได้รับ จะถูกแปลงค่าเป็นตัวเลข เช่น ยิ่งแสงน้อยเท่าไหร่ ยิ่งแสดงค่าน้อยลงเท่านั้น ซึ่งในตัวอย่างนี้เราสร้าง Sketch Code เพื่อให้บอร์ด Arduino ควบคุมและแสดงผลการทำงานของ Photoresistor Sensor Module

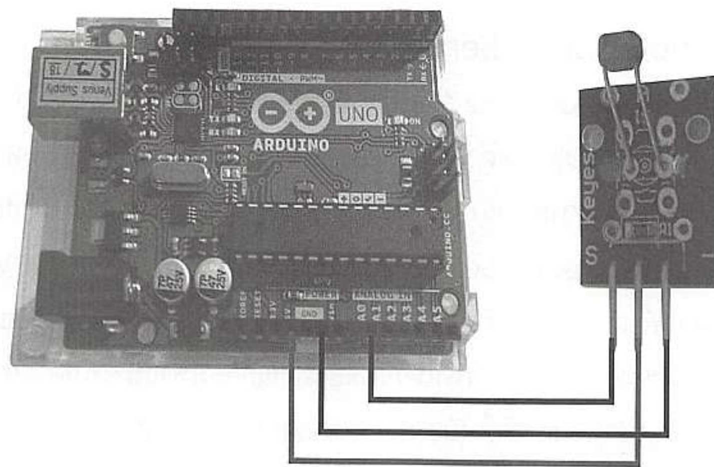
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Photoresistor Sensor Module
3. Jumper Wire
4. Breadboard

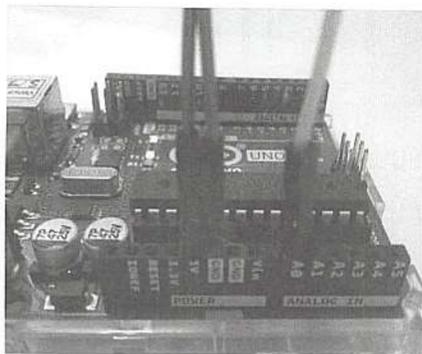


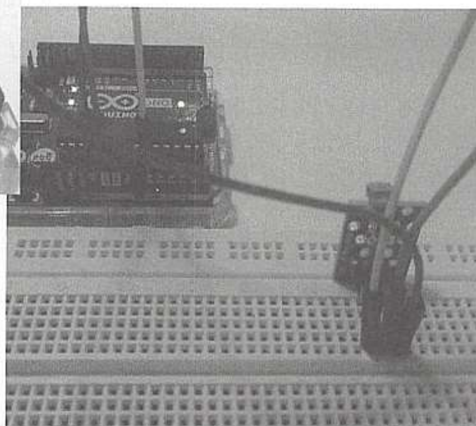
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด AO บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



Photoresistor Sensor Module จะส่งสัญญาณอะนาล็อก ซึ่งเป็นค่าของแสงไฟ LED ตรวจจับได้ไปที่บอร์ด Arduino จากนั้นบอร์ดจะทำการประมวลผลและแปรสัญญาณเป็นค่าตัวเลข ซึ่งเราสามารถดูค่าที่แสดงนั้นได้ ด้วยการคลิกเมนู Monitor

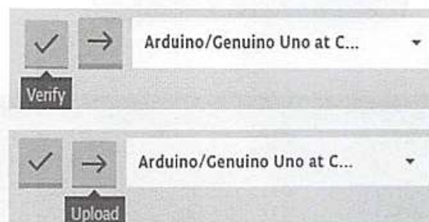




Sketch Code

```
int sensorPin = A0;  // select the analog input pin for the photoresistor
void setup() {
Serial.begin(9600);
}
void loop() {
 Serial.println(analogRead(sensorPin));
 delay(200);
 }
```

การ Verify (Compile) และการ Upload



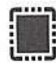
Success: Done verifying

Success: Done uploading

เมื่อระบบทำการ Upload Sketch Code ไปยังบอร์ด Arduino Success แล้ว Photoresistor Sensor Module จะรับสัญญาณจาก LDR จากนั้นส่งสัญญาณอะนาล็อก (AO) ไปยังบอร์ด Arduino และจะแสดงค่าตัวเลขใน Serial Monitor

ค่าของแสงที่เซ็นเซอร์วัดได้

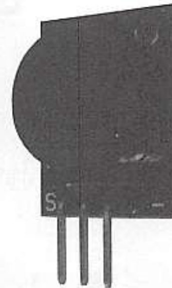
ลองลดแสงให้น้อยลง ค่าแสงจะน้อยลงตามทันที

 Passive Buzzer Module

Passive Buzzer Module ก็คือ โมดูลส่งเสียงเตือนคล้ายกับ Buzzer Module ในโปรเจ็กต์ก่อนหน้านี้ เพียงแต่โปรเจ็กต์นี้พิเศษกว่าตรงที่ Passive Buzzer Module ส่งเสียงที่มีความถี่สูง อีกทั้งยังสามารถกำหนดความถี่เสียงที่เราต้องการได้อีกด้วย โดยการทดลองนี้ เราจะเขียน Sketch Code ให้ Passive Buzzer Module ส่งเสียง Melody ไม่ใช่สัญญาณเตือน ติ๊ด ติ๊ด ธรรมดาๆ

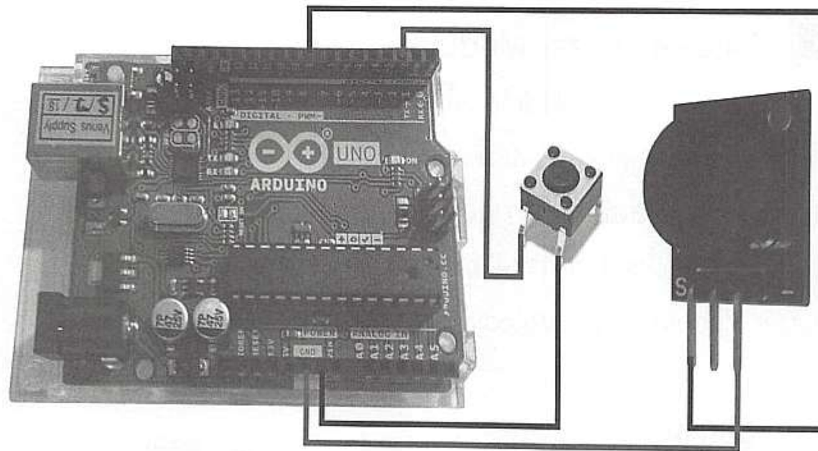
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Passive Buzzer Module
3. Push Button Switch
4. Jumper Wire
5. Breadboard

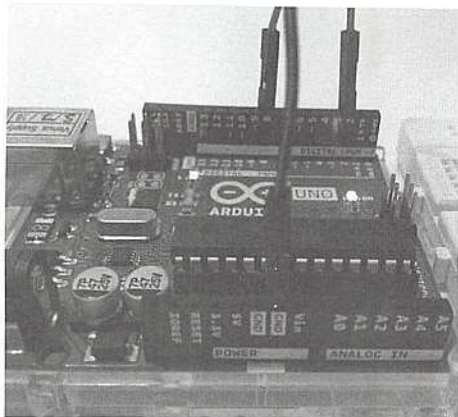


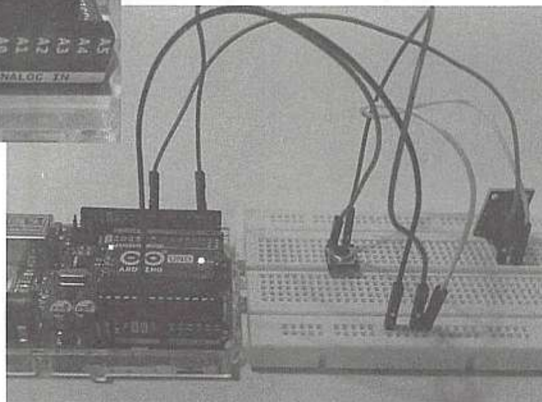
การต่อวงจร

- ขา S ของโมดูลต่อกับหมุด D8 บนบอร์ด Arduino
- ขา - ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- Push Button Switch - ต่อกับหมุด GND บนบอร์ด Arduino
- Push Button Switch + ต่อกับหมุด D2 บนบอร์ด Arduino



เมื่อกดปุ่มสัญญาณจะถูกส่งไปที่หมุด D2 และบอร์ดจะส่งสัญญาณไปที่หมุด D8 เพื่อส่งสัญญาณไปที่ขา S ของ Passive Buzzer Module





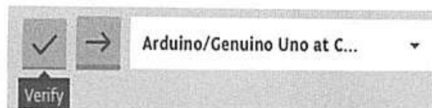
Sketch Code

```
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
```

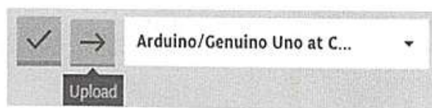
```
// notes in the melody: from guitar study by Sor
int melody[ ] = {
 NOTE_G3, NOTE_G3, NOTE_E3, NOTE_C4, NOTE_B3, NOTE_B3, NOTE_
A3, NOTE_D4, NOTE_C4, NOTE_C4, NOTE_B3, NOTE_G4, NOTE_F4,
 NOTE_F4, NOTE_E4, NOTE_C4, NOTE_A3, NOTE_G3, NOTE_E3, NOTE_
C4, NOTE_B3, NOTE_B3, NOTE_A3, NOTE_D4, NOTE_C4, NOTE_C4,
 NOTE_B3, NOTE_E4, NOTE_D4, NOTE_C4
};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 2
};
void setup() {
 pinMode(2, INPUT_PULLUP); // push-button switch on pin 2, use
internal pull-up resistor
 PlayTune();        // play tune at start-up
}
void loop() {
 // play tune again if button on digital pin 2 is pressed
 if (!digitalRead(2)) {
  PlayTune();
 }
}
// a function that plays the tune
void PlayTune()
{
```

```
// iterate over the notes of the melody:
for (int thisNote = 0; thisNote < 30; thisNote++) {
  // to calculate the note duration, take one second
  // divided by the note type.
  //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
  int noteDuration = 1500 / noteDurations[thisNote];
  tone(8, melody[thisNote], noteDuration);
  // to distinguish the notes, set a minimum time between them.
  // the note's duration + 30% seems to work well:
  int pauseBetweenNotes = noteDuration * 1.30;
  delay(pauseBetweenNotes);
  // stop the tone playing:
  noTone(8);
 }
}
```

การ Verify (Compile) และการ Upload

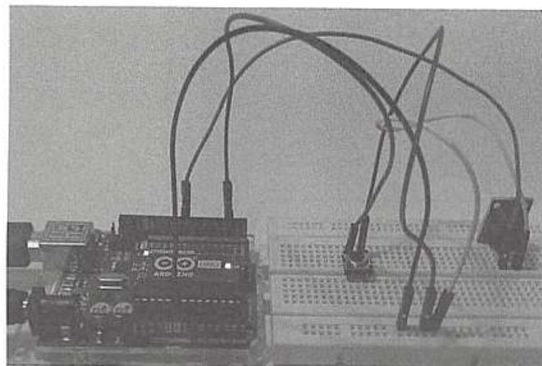


Success: Done verifying

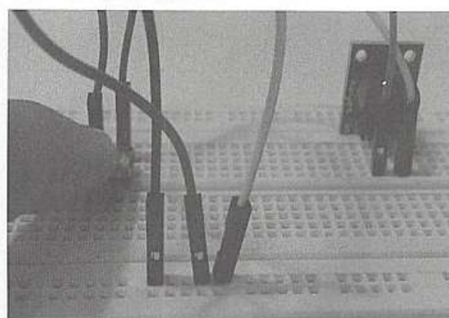


Success: Done uploading

เมื่อทำการ Upload Sketch Code ไปยังบอร์ด Arduino แล้ว เมื่อกดปุ่ม 1 ครั้ง Melody จะดังขึ้น 1 รอบ จนกว่าจะมีการกดปุ่มอีกครั้ง





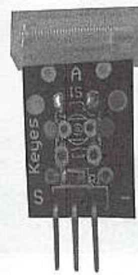


 Knock Sensor Module

Knock Sensor Module คือ เซ็นเซอร์ที่มีคุณสมบัติในการตรวจจับแรงสั่นสะเทือนแบบทันทีทันใด Knock ก็แปลตามตัวเลย คือ การเคาะ โดยเราจะให้ Knock Sensor Module ส่งสัญญาณไปที่บอร์ด Arduino เพื่อแสดงค่าเป็นตัวเลขบน Monitor

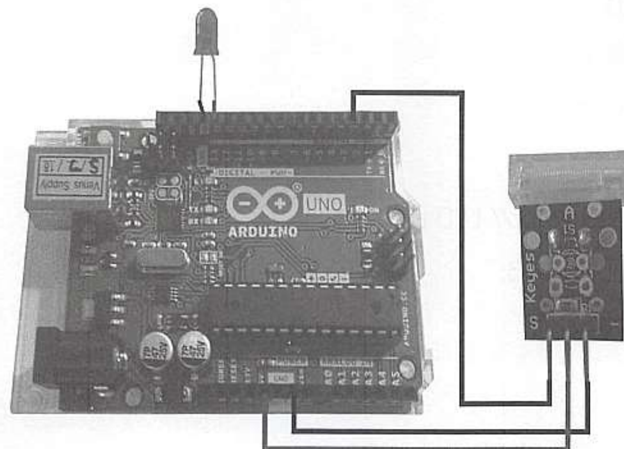
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Knock Sensor Module
3. LED
4. Jumper Wire

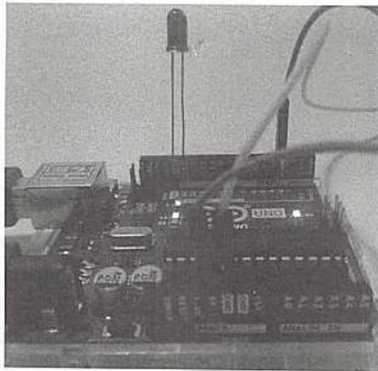


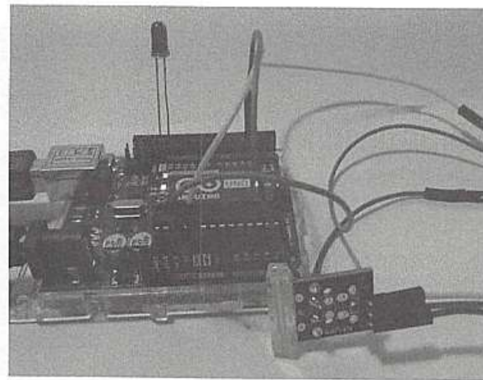
การต่อวงจร

- ขา S ของเซ็นเซอร์ต่อกับหมุด ~3 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด 13 บนบอร์ด Arduino
- ขา LED - ไปต่อกับหมุด GND บนบอร์ด Arduino



เมื่อการเคาะเซ็นเซอร์เกิดขึ้น สปริงในหลอดแก้วจะสั่นและโมดูลจะส่งสัญญาณไปที่ ~3 จากนั้นบอร์ดจะทำให้ไฟ LED ติดกะพริบ และแสดงค่าตัวเลขใน Serial Monitor





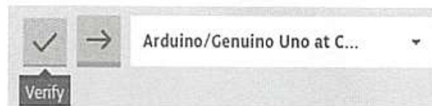
Sketch Code

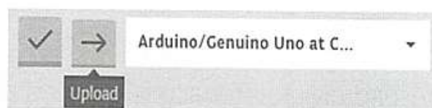
```
// Arduino code is available to download - link below the video
/*
Arduino   Knock/Shock Sensor KY-031
 GND        -
 5V         +5V
 D3         S
*/
int Led = 13;// LED pin
int Shock = 3;
int val;
```



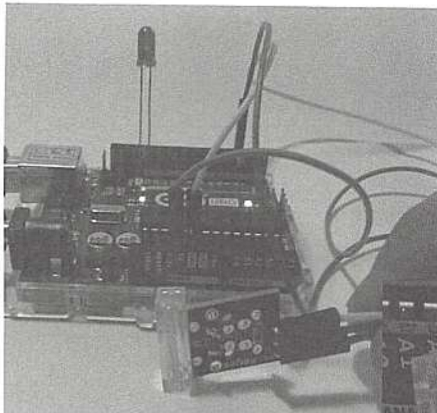
```
void setup () {
 Serial.begin(9600);
 pinMode(Led, OUTPUT);
 pinMode(Shock, INPUT);
}
void loop () {
 val = digitalRead (Shock);// read digital interface is assigned a
value of 3 val
 Serial.println(val);
 if (val == HIGH) {// When the percussion when the sensor detects
a signal, LED flashes
 digitalWrite(Led, LOW);
 }
 else {
 digitalWrite(Led, HIGH);
 }
}
```

การ Verify (Compile) และการ Upload

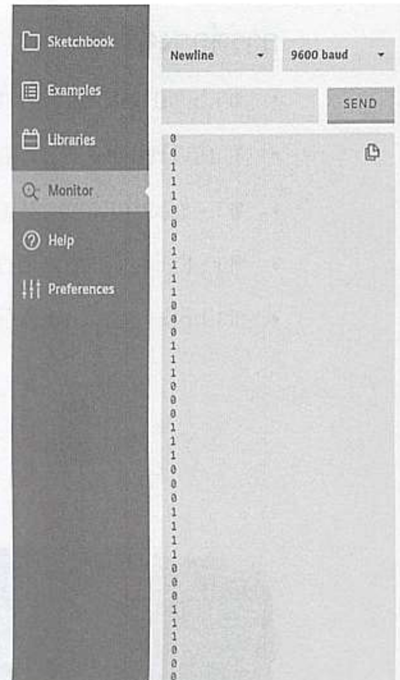




เมื่อได้ทำการ Upload Sketch Code ไปยังบอร์ด Arduino เสร็จเรียบร้อยแล้ว หากเคาะเซ็นเซอร์ (เบาๆ) ไฟ LED ที่รับสัญญาณจากหมุด 13 ของบอร์ด Arduino จะติดกะพริบ







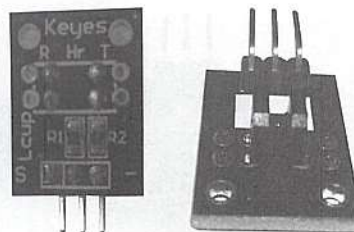
หากเลือกที่เมนู Monitor ด้านซ้ายมือ จะเห็นค่าตัวเลขที่ระบบแสดง 0 คือ ไม่มีการเคาะเซ็นเซอร์ 1 คือ มีการเคาะเซ็นเซอร์เกิดขึ้น (การสั่นสะเทือนเกิดขึ้น)

 Light Blocking Sensor

Light Blocking Sensor คือ เซ็นเซอร์ที่มีคุณสมบัติตรวจจับวัตถุที่ผ่านตัดหน้าเซ็นเซอร์ โดยอาศัยหลักการ คือ บนตัวเซ็นเซอร์จะมีการติดตั้งตัวส่งแสง (Emitter) และตัวรับแสง (Receiver) ซึ่งโดยปกติหน้าเซ็นเซอร์จะส่งและรับแสงได้ตลอดเวลา แต่เมื่อใดที่มีวัตถุผ่านหน้าเซ็นเซอร์ จะทำให้ตัวรับแสงไม่สามารถรับแสงได้ เซ็นเซอร์จึงจะส่งสัญญาณออกไป

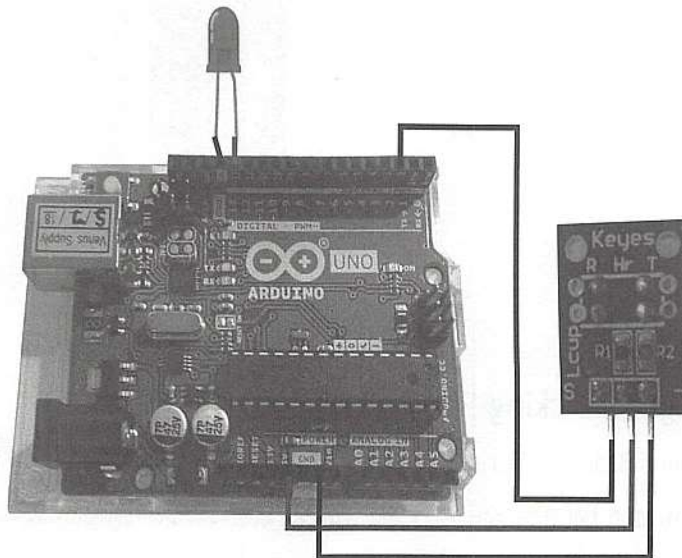
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Light Blocking Sensor
3. LED
4. Jumper Wire

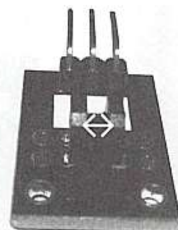


การต่อวงจร

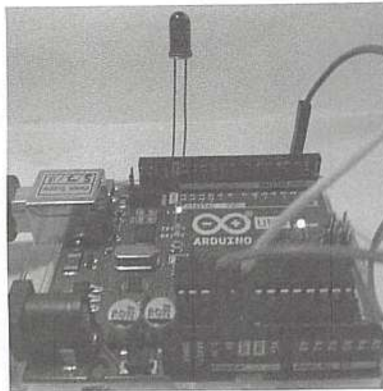
- ขา S ของเซ็นเซอร์ต่อกับหมุด D2 บนบอร์ด
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino
- ขา LED + ต่อกับหมุด 13 บนบอร์ด Arduino
- ขา LED - ต่อกับหมุด GND บนบอร์ด Arduino

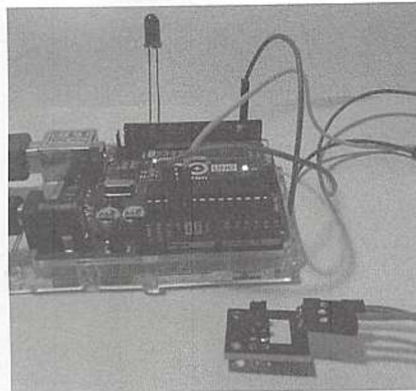


เมื่อมีวัตถุทึบแสงผ่านตัดหน้าเซ็นเซอร์ โมดูลจะส่งสัญญาณไปที่หมุด 2 บนบอร์ด Arduino จากนั้นจะส่งสัญญาณไปที่หมุด 13 ทำให้ไฟ LED ติด



หน้าเซ็นเซอร์สำหรับส่ง-รับแสง



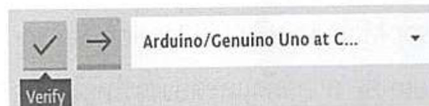


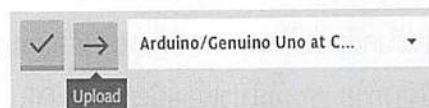
Sketch Code

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin
// variables will change:
int buttonState = 0; // variable for reading the pushbutton status
void setup() {
// initialize the LED pin as an output:
pinMode(ledPin, OUTPUT);
// initialize the pushbutton pin as an input:
pinMode(buttonPin, INPUT);
}
void loop(){
// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);
// check if the pushbutton is pressed.
```

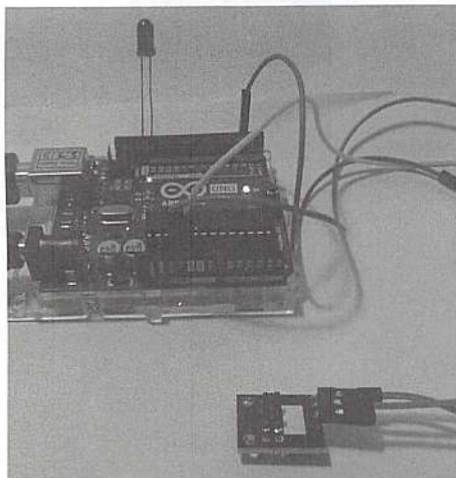
```
// if it is, the buttonState is HIGH:
if (buttonState == HIGH) {
// turn LED on:
digitalWrite(ledPin, HIGH);
}
else {
// turn LED off:
digitalWrite(ledPin, LOW);
}
}
```

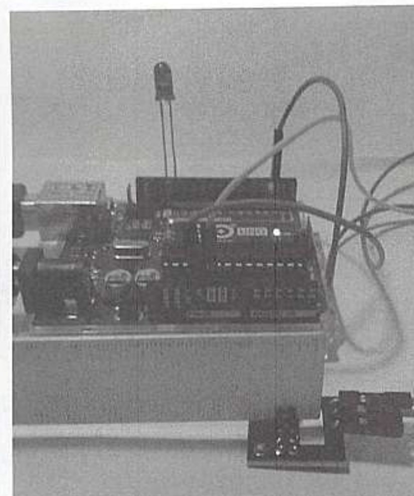
การ Verify (Compile) และการ Upload





หลังจากที่ Upload Sketch Code ไปยังบอร์ด Arduino Success แล้ว หากนำวัตถุทึบแสงผ่านหน้าเซ็นเซอร์ ไฟ LED ที่รับสัญญาณจากหมุด 13 บนบอร์ด Arduino จะติด



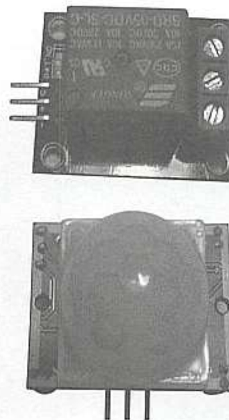


 Relay และ Motion Sensor Module

Relay and Motion Sensor Module คือ โปรเจ็กต์ที่นำ Relay Module ใช้งานร่วมกับ Motion Sensor Module หรือ PIR Motion Sensor แน่นอนอย่างที่รู้ๆ กันอยู่ Relay Module คือ สวิตช์ตัดและต่อวงจรโดยใช้หลักการของแม่เหล็กไฟฟ้า จะทำงานเมื่อมีการจ่ายไฟไปตามกำหนดเท่านั้น ส่วน PIR Motion Sensor คือ เซ็นเซอร์ที่ใช้ตรวจจับคลื่นรังสีอินฟราเรด ที่แพร่รังสีออกมาจากตัวมนุษย์ หรือสิ่งมีชีวิตที่มีอุณหภูมิร่างกาย เช่นนั้นโปรเจ็กต์นี้เราจะใช้ PIR Motion Sensor ตรวจจับรังสีจากร่างกายของเรา แล้วส่งสัญญาณไปที่บอร์ด Arduino เพื่อให้บอร์ดส่งสัญญาณไปที่ Relay Module ให้กระแสไฟฟ้าไหลเข้าซึ่งจะส่งผลทำให้โคมไฟติดสว่าง

อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Relay Module
3. PIR Motion Sensor
4. Jumper Wire
5. Bread Board



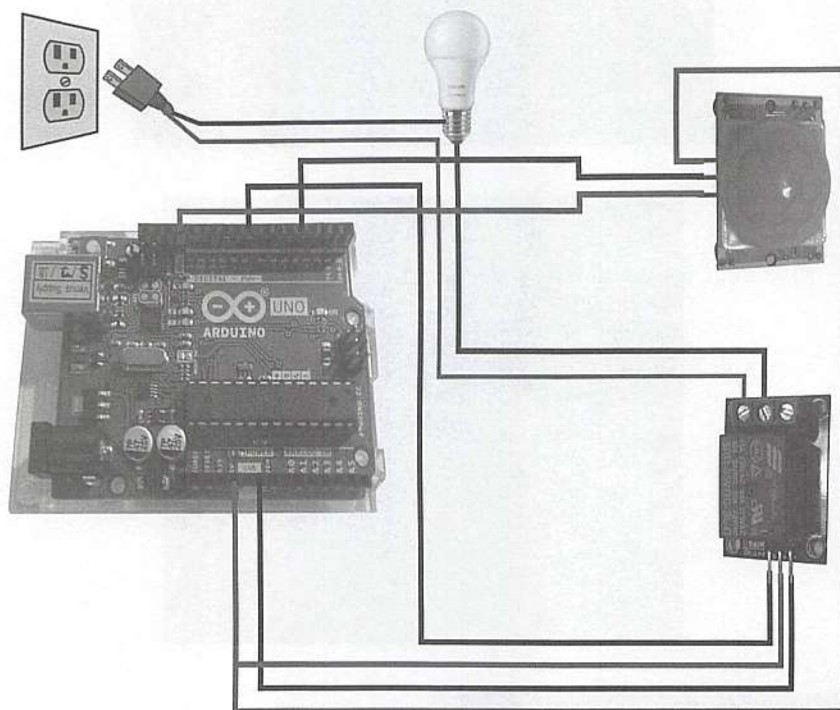
การต่อวงจร

Relay Module

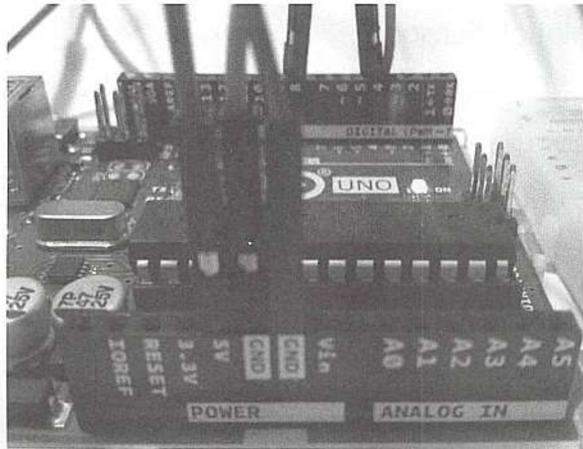
- สาย L ของโคมไฟต่อกับขา COM ของ Relay Module
- ขา NC ของ Relay Module ต่อกับโคมไฟ
- สาย N ของโคมไฟต่อกับปลั๊กไฟ ACV
- ขา S ของ Relay Module ต่อกับหมุด D8 บนบอร์ด Arduino
- ขา + ของ Relay Module ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของ Relay Module ต่อกับหมุด GND บนบอร์ด Arduino

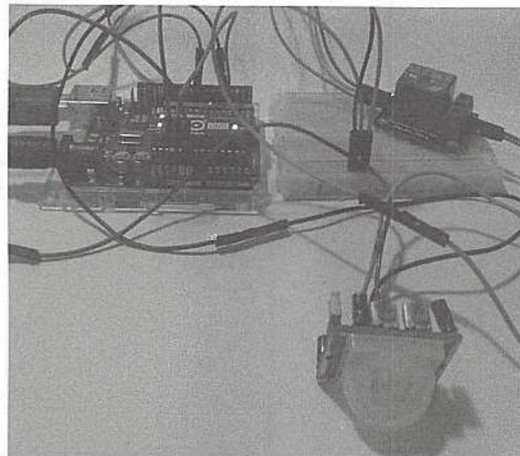
PIR Motion Sensor

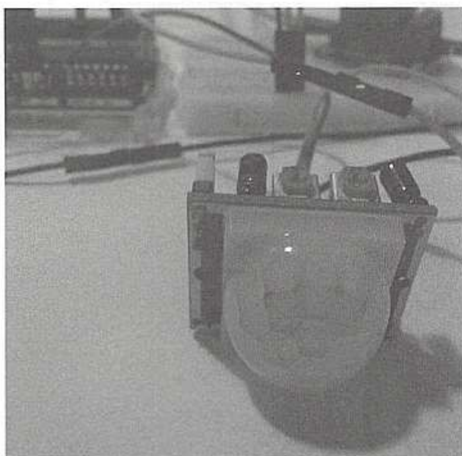
- ขา VCC ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา OUT ของเซ็นเซอร์ต่อกับหมุด D4 บนบอร์ด Arduino
- ขา GND ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อ PIR Motion Sensor ตรวจจับรังสีอินฟราเรดจากมือของเราได้ จะส่งสัญญาณไปที่หมุด 4 ของบอร์ด Arduino จากนั้นบอร์ดจะส่งสัญญาณไปที่ Relay Module เพื่อปล่อยให้แรงดันไฟฟ้าไหลเข้า Relay Module จนเกิดสนามแม่เหล็กทำให้หน้าสัมผัสของ Relay Module แตะกัน ครบวงจรของโคมไฟ "โคมไฟติด"


UNO
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
POWER
ANALOG IN






SRD-05VDC-SL-C
SONGLE
Keyes_SR1y

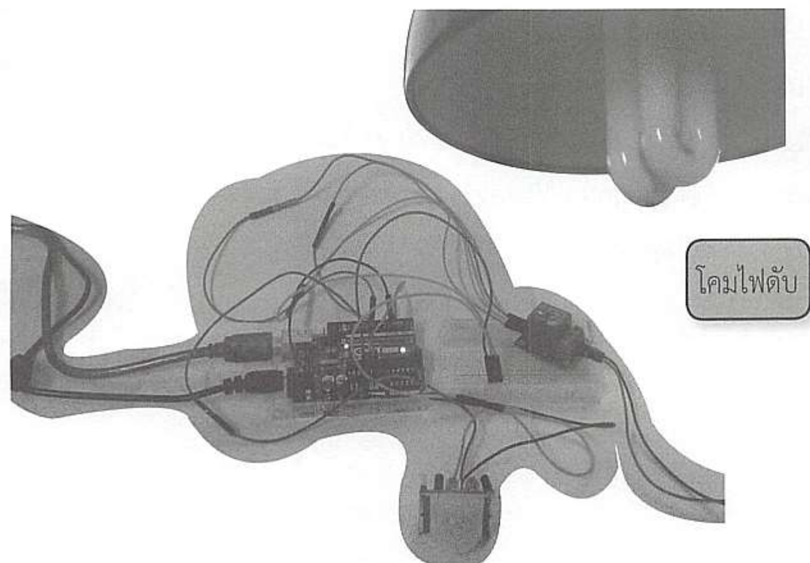
Sketch Code

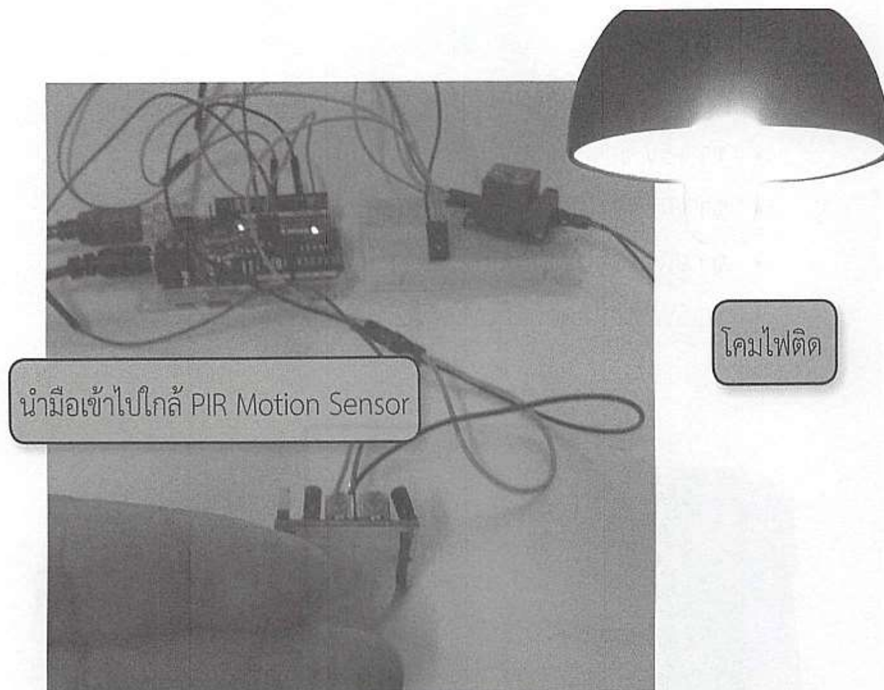
```
int irmotionPin = 4; // Pin of IR Motion Sensor
int relayPin = 8; // Pin of Relay Module
void setup(){
 Serial.begin(9600);
 pinMode(relayPin, OUTPUT); // Set Pin connected to Relay as an
OUTPUT
 digitalWrite(relayPin, LOW); // Set Pin to LOW to turn Relay OFF
}
void loop(){
 while (digitalRead(irmotionPin) == HIGH) { // If Motion detected
   digitalWrite(relayPin, HIGH); // Turn Relay ON
   Serial.println("Relay is ON");
   delay(500);
}
   digitalWrite(relayPin, LOW); // Turn Relay OFF
   Serial.println("Relay is OFF");
   delay(500);
}
```

การ Verify (Compile) และการ Upload



เมื่อนำมือเข้าใกล้ PIR Motion Sensor เซ็นเซอร์จะตรวจจับรังสีความร้อนหรือรังสีอินฟราเรดได้ และส่งสัญญาณไปที่บอร์ด Arduino จากนั้นบอร์ดจะส่งสัญญาณไปที่ Relay Module เพื่อปล่อยให้กระแสไฟไหลเข้าจึงเกิดเป็นสนามแม่เหล็ก ทำให้สวิตช์เปิด โคมไฟติด 500 มิลลิวินาที แล้วดับ และหาก PIR Motion Sensor ยังคงตรวจจับรังสีความร้อนได้อยู่ โคมไฟจะดับ 500 มิลลิวินาที โคมไฟจะติดสว่างขึ้นอีกครั้ง



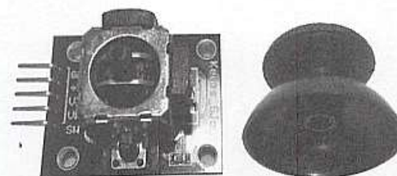


 Analog Joystick

คงจะเป็นที่รู้จักกันดีมานานแล้ว Joystick เครื่องมือเล่นเกมยอดฮิตที่มีมาตั้งแต่ 20 ปีก่อน ซึ่งในปัจจุบันได้มีการพัฒนาให้ Joystick สามารถทำอะไรได้หลายต่อหลายอย่างมากยิ่งขึ้น และมีอุปกรณ์มากมายที่สามารถรองรับการใช้งาน Joystick สำหรับโปรเจ็กต์นี้ เราจะมาลองต่อวงจรของ Joystick Module ซึ่งเป็นสัญญาณอะนาล็อก กับบอร์ด Arduino โดยการเขียน Sketch Code เพื่อ Test ระบบการทำงานของ Sketch Code

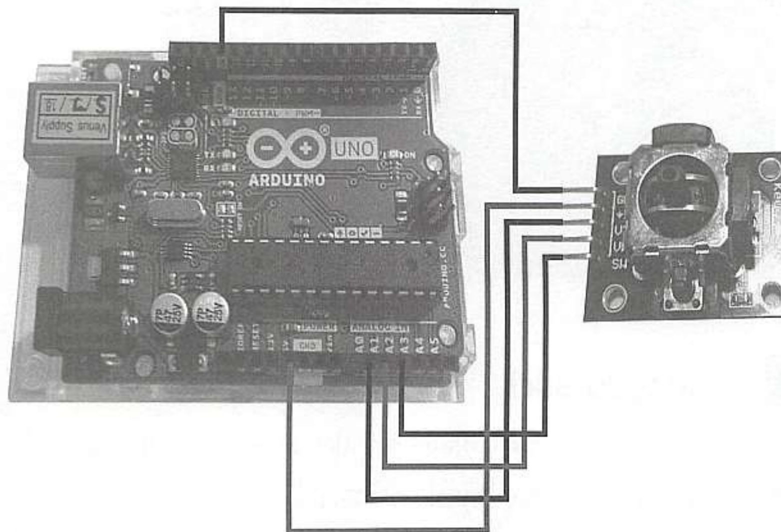
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Joystick Module
3. Jumper Wire

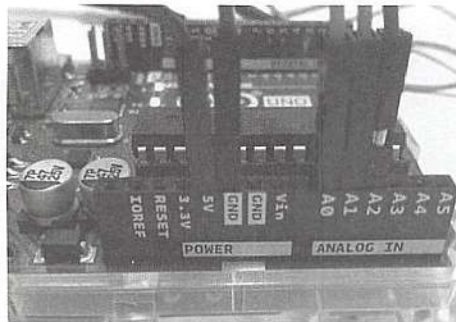


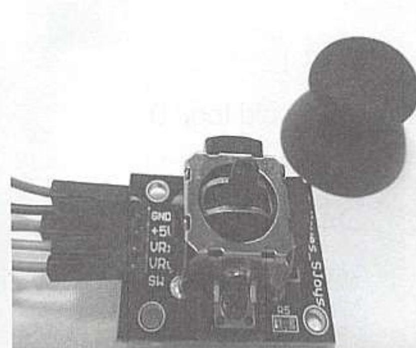
การต่อวงจร

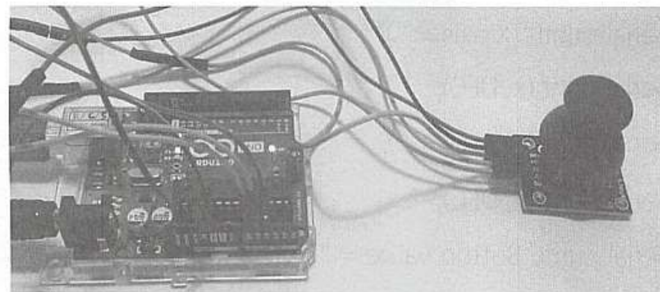
- ขา GND ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา +5V ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา VRx ของโมดูลต่อกับหมุด AO บนบอร์ด Arduino
- ขา VRy ของโมดูลต่อกับหมุด A1 บนบอร์ด Arduino
- ขา SW ของโมดูลต่อกับหมุด A2 บนบอร์ด Arduino



เมื่อเลื่อนหรือกด Joystick โมดูลจะส่งสัญญาณอะนาล็อกไปที่หมุด AO, A1 และ A2 ไปยังบอร์ด Arduino เพื่อทำการประมวลผลและแสดงค่าใน Serial Monitor







Sketch Code

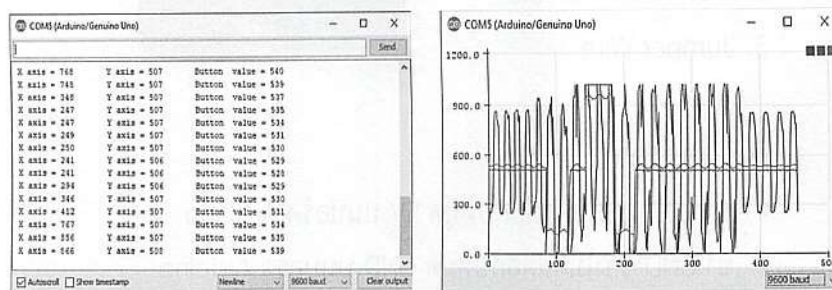
```
//
int JoyStick_X = A0; // x
int JoyStick_Y = A1; // y
int JoyStick_Z = A2; // button
void setup ()
{
 pinMode (JoyStick_X, INPUT);
 pinMode (JoyStick_Y, INPUT);
 pinMode (JoyStick_Z, INPUT);
 Serial.begin (9600); // 9600 bps
```

```
}
void loop ()
{
 int x, y, z;
  x = analogRead (JoyStick_X);
 y = analogRead (JoyStick_Y);
 z = analogRead (JoyStick_Z);
 Serial.print(" X axis = ");
 Serial.print (x, DEC);
 Serial.print(" Y axis = ");
 Serial.print (y, DEC);
 Serial.print("Button value = ");
 Serial.print (z, DEC);
 delay(100);
 if(z == 0){
 Serial.print(" BUTTON is PRESSED");
 delay(1000);
 }
 Serial.println(); // start a new line
 delay (100);
}
```

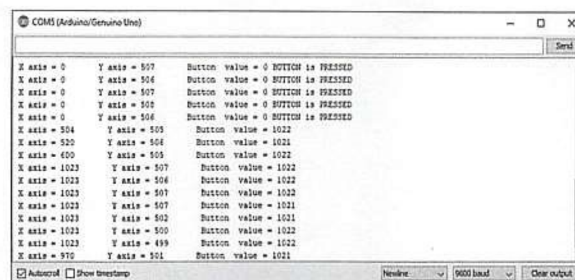
การ Verify (Compile) และการ Upload



ให้คลิกเลือกที่ Serial Monitor เพื่อดูค่าต่างๆ ของคันบังคับของ Joystick หรือเลือกดูที่ Serial Plotter ใน Tools เพื่อดูค่าที่แสดงเป็นกราฟเส้น



เมื่อเรากดคันบังคับ (ไม่โยกขึ้นลงซ้ายขวา แต่กดลงตรงๆ) Serial Plotter จะแสดงผลเป็น Button Value = 1021 - 1022

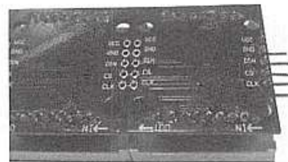


 Dot Matrix Module

Dot Matrix Module 4-in-1 คือ โมดูลไฟ LED 4 ชุด ที่ต่ออนุกรมกัน โดยเราสามารถใช้งาน Dot Matrix Module ได้หลายโปรเจ็กต์ตามต้องการ เพียงแต่พิมพ์ Sketch Code ที่เราต้องการให้ปรากฏบนชุดไฟ LED โดยโปรเจ็กต์นี้จะเป็นการใช้งาน Dot Matrix Module แบบง่ายๆ ซึ่งควบคุมการทำงานด้วยบอร์ด Arduino

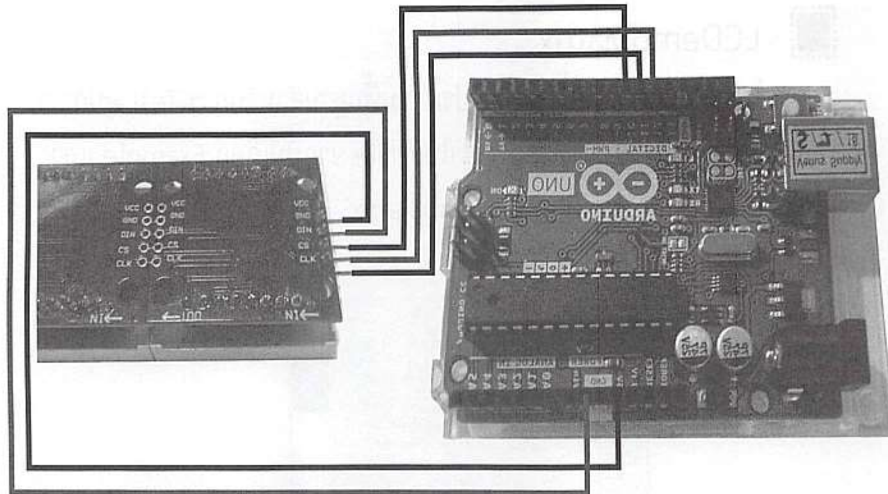
อุปกรณ์ที่ใช้

1. Arduino Uno R3
2. Dot Matrix Module
3. Jumper Wire

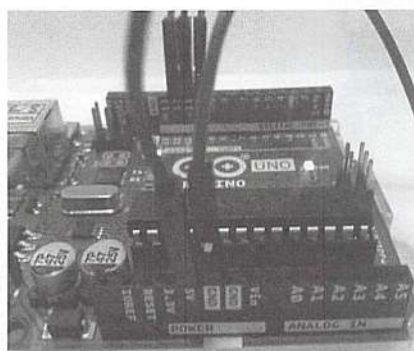


การต่อวงจร

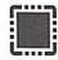
- ขา VCC ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา GND ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino
- ขา DIN GND ของโมดูลต่อกับหมุด ~10 บนบอร์ด Arduino
- ขา CS (Chip Select) ของโมดูลต่อกับหมุด D12 บนบอร์ด Arduino
- ขา CLK (Clock) ของโมดูลต่อกับหมุด ~11 บนบอร์ด Arduino



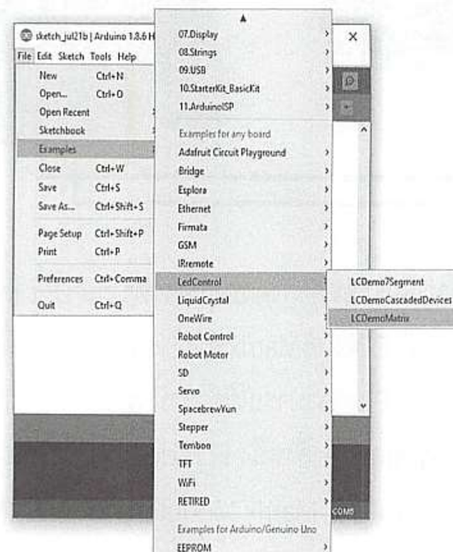
การทดสอบโปรแกรมชุดคำสั่งควบคุมการทำงาน Dot Matrix Module จะมีอยู่ด้วยกัน 2 โหมด LCDemoMatrix คือ Sketch Code พื้นฐานที่ติดมากับโปรแกรม Arduino IDE เราสามารถใช้คำสั่งนี้ เพื่อลอง Test การทำงานของอุปกรณ์ได้เลยทันที ส่วนอีกชุดคำสั่งหนึ่ง LED demo loop คือ Sketch Code การทดสอบการทำงานของ Dot Matrix Module ให้ไฟ LED โดยจะให้ไฟ LED ติด ตามจังหวะวนซ้ำไปเรื่อยๆ





 LCDemoMatrix

โปรแกรมชุดคำสั่งนี้ จะเป็นโปรแกรมพื้นฐานสำหรับการ Test อุปกรณ์บนโปรแกรม Arduino IDE โดยให้เราไปที่ File จากนั้นเลือก Example แล้วเลือก LedControl แล้วเลือก LCDemoMatrix

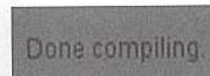




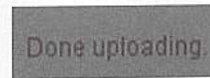
ระบบโปรแกรม Arduino IDE จะเปิดไฟล์ Sketch Code ของ LCDemoMatrix ขึ้นมาทันที

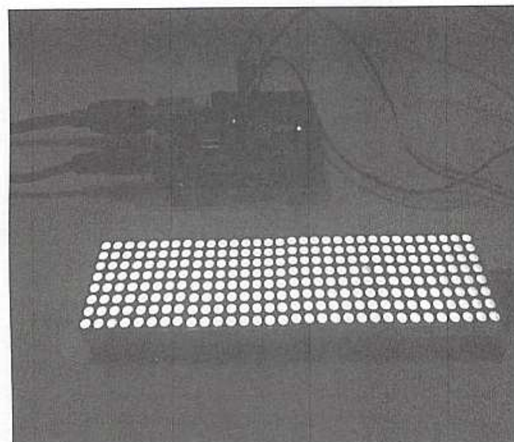
โดยเราสามารถ Verify และ Upload Sketch Code นี้ได้ทันที เมื่อต่อวงจรเรียบร้อยแล้ว ไฟ LED บน Dot Matrix Module จะเปล่งแสงตามจังหวะและเวลาที่ได้กำหนดไว้ใน Sketch Code

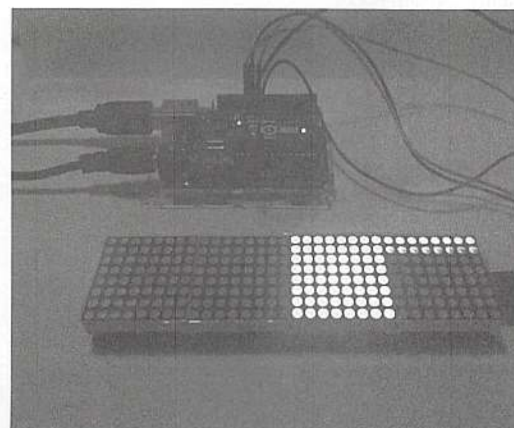












Sketch Code LED demo loop

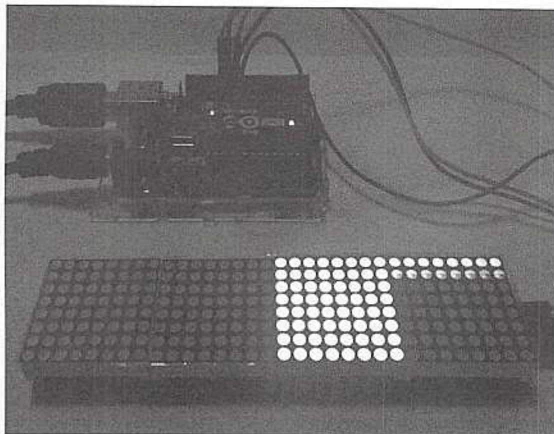
```
// Inladen van de ledcontrol bibliotheek.
#include "LedControl.h"
LedControl lc=LedControl(12, 11, 10, 1); // DIN, CLK, CS, NRDEV
//Een variabele voor het wachten voordat we het display updaten.
unsigned long delaytime = 50;
void setup() {
  // Haal het aantal apparaten op dat we hebben "gecreëerd" met
Ledcontrol.
  int devices=lc.getDeviceCount();
  // Alle apparaten initialiseren (in een loop).
  for(int address=0;address<devices;address++) {
    // De MAX72XX IC is in slaapstand modus bij opstarten.
    lc.shutdown(address,false);
    // Zet de helderheid op een medium niveau.
    lc.setIntensity(address,8);
    // Maak de dot matrix leeg (clear display).
    lc.clearDisplay(address);
  }
}
void loop() {
  // Lees het aantal apparaten uit.
  int devices=lc.getDeviceCount();
  // Laat de ledjes stuk voor stuk branden.
  for(int row=0;row<8;row++) {
```

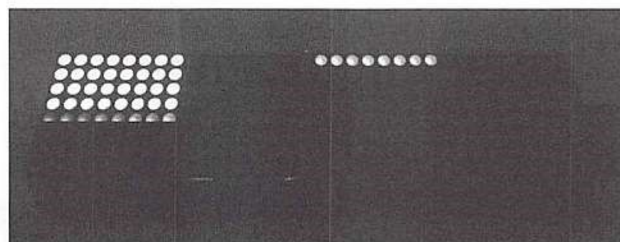
```
    for(int col=0;col<8;col++) {
      for(int address=0;address<devices;address++) {
        lc.setLed(address,row,col,true); delay(delaytime);
      }
    }
  }
  // Zet de ledjes stuk voor stuk uit.
  for(int row=0;row<8;row++) {
    for(int col=0;col<8;col++) {
      for(int address=0;address<devices;address++) {
        lc.setLed(address,row,col,false); delay(delaytime);
      }
    }
  }
}
```

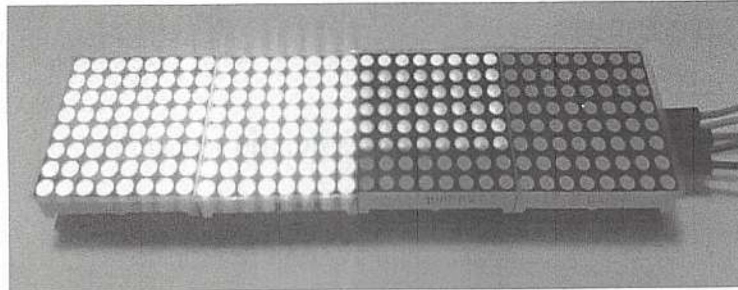
การ Verify (Compile) และการ Upload



เมื่อระบบ Upload Sketch Code "Done" แล้ว บอร์ด Arduino จะส่งสัญญาณไปยัง Dot Matrix Module ไฟ LED ทุกดวง จะพร้อมรับคำสั่งติดและดับตามโปรแกรม







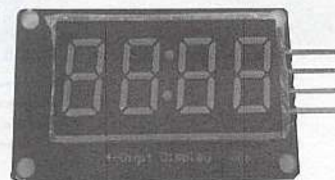
 Temperature Humidity Sensor และ LED 4-Digit Display Module

สำหรับโปรเจ็กต์นี้ จะเป็นการตรวจวัดค่าอุณหภูมิและค่าความชื้นในอากาศโดยใช้ Temperature Humidity Sensor Module และเขียน Sketch Code ให้แสดงค่าที่วัดได้ไปที่ LED 4-Digit Display Module ซึ่งเราจะสามารถทราบค่าอุณหภูมิและค่าความชื้นในอากาศได้ทันที ด้วยตัวเลขที่ปรากฏบน LED 4-Digit Display Module ภายใต้คำสั่งของบอร์ด Arduino

อุปกรณ์ที่ใช้



1. Arduino Uno R3
2. Temperature Humidity Sensor Module
3. LED 4-Digit Display Module
4. Jumper Wire
5. Breadboard



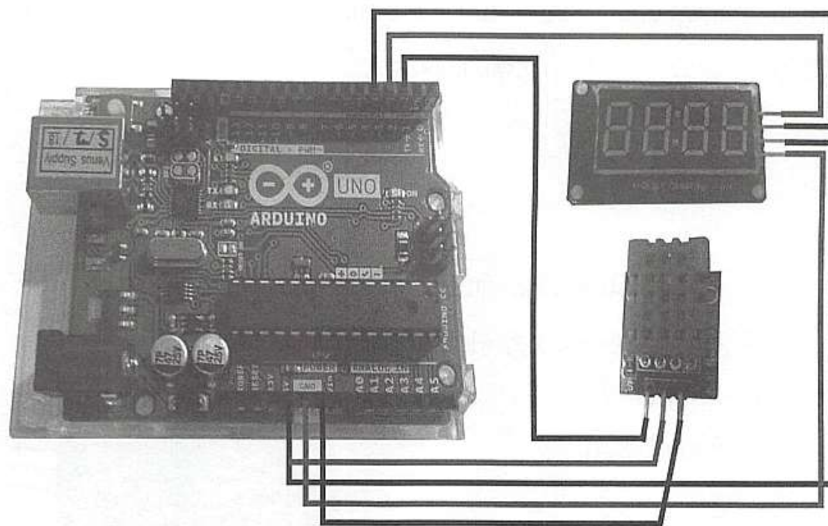
การต่อวงจร

LED 4-Digit Display Module

- ขา CLK ของโมดูลต่อกับหมุด ~3 บนบอร์ด Arduino
- ขา DIO ของโมดูลต่อกับหมุด D4 บนบอร์ด Arduino
- ขา VCC ของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา GND ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino

Temperature Humidity Sensor Module

- ขา S ของเซ็นเซอร์ต่อกับหมุด D2 บนบอร์ด Arduino
- ขากลางของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino



เมื่อ Temperature Humidity Sensor Module ตรวจวัดค่าอุณหภูมิและค่าความชื้นได้ จะส่งสัญญาณไปที่บอร์ด Arduino จากนั้นบอร์ดจะส่งสัญญาณไปที่ LED 4-Digit Display Module

Libraries

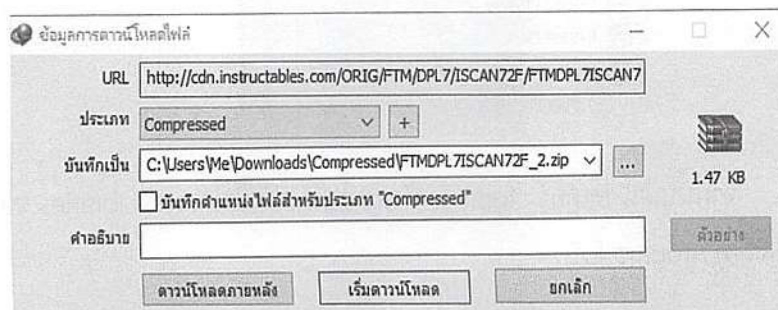
โปรเจ็กต์นี้จะต้องทำการดาวน์โหลด Libraries แล้วนำโฟลเดอร์ Libraries ของ Sketch Code ไปเก็บไว้ในโฟลเดอร์ Libraries ของโปรแกรม Arduino IDE โดยการดาวน์โหลด Libraries จำเป็นจะต้องดาวน์โหลดทั้ง 2 โมดูล

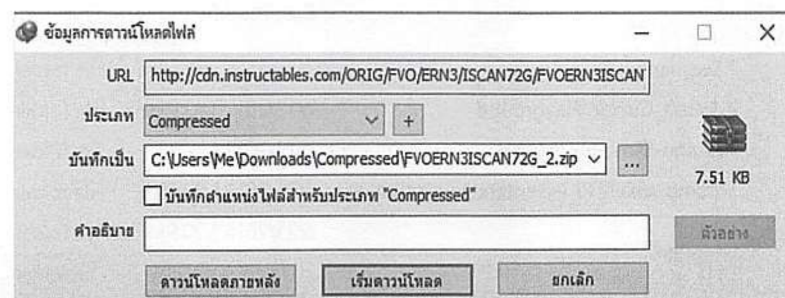
LED 4-Digit Display Module (TM1637) ดาวน์โหลดที่ Link นี้

http://www.instructables.com/files/orig/FVO/ERN3/ISCAN72G/FVOERN3ISCAN72G.zip

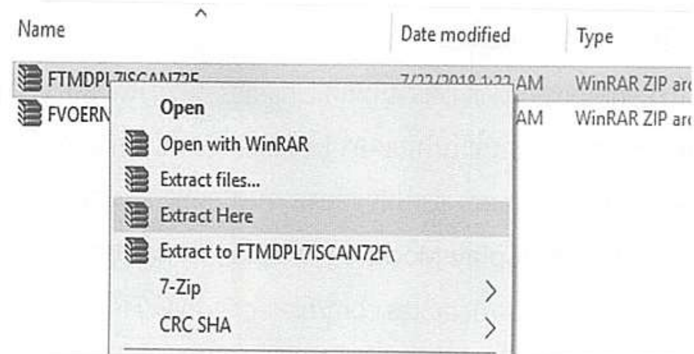
Temperature Humidity Sensor Module (DHT11) ดาวน์โหลดที่ Link นี้

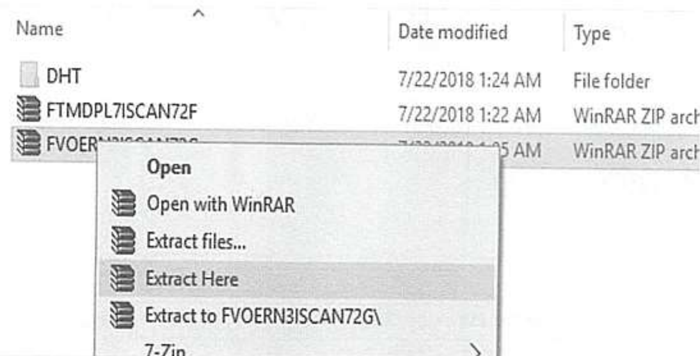
http://www.instructables.com/files/orig/FTM/DPL7/ISCAN72F/FTMDPL7ISCAN72F.zip





เมื่อดาวน์โหลดไฟล์ Libraries มาเก็บไว้ในหน่วยความจำคอมพิวเตอร์แล้ว ให้ทำการแตกไฟล์ จะได้โฟลเดอร์ Libraries ของ Sketch Code ทั้ง 2 โมดูล





จากนั้นนำโฟลเดอร์ Libraries ทั้งสองไปเก็บไว้ที่โฟลเดอร์ Libraries ของโปรแกรม Arduino IDE

Name	Date modified	Type
7 Segment	7/21/2018 10:11 PM	File folder
Adafruit_Circuit_Playground	7/18/2017 11:21 PM	File folder
Arduino-IRremote-master	7/12/2018 1:36 AM	File folder
arduino-max7219-led-interactive-scrollin...	7/21/2018 8:32 PM	File folder
Bridge	8/25/2016 5:20 PM	File folder
DHT	7/22/2018 1:03 AM	File folder
DigitalTube	7/22/2018 1:05 AM	File folder
Esplora	5/20/2015 5:10 PM	File folder

Sketch Code

```
#include <dht.h>
#include "TM1637.h"
//{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
//0~9,A,b,C,d,E,F
#define dht_pin 2 // Pin sensor is connected to
#define CLK 3//Pins for TM1637
#define DIO 4
TM1637 tm1637(CLK,DIO);
dht DHT;
void setup(){
 tm1637.init();
 tm1637.set(BRIGHT_DARKEST);
 //BRIGHT_TYPICAL = 2,BRIGHT_DARKEST = 0,BRIGHTEST = 7;
  delay(500);
 }
void loop(){
 DHT.read11(dht_pin);
 int temp = DHT.temperature;
 int humidity = DHT.humidity;
 int digitoneT = temp / 10;
 int digittwoT = temp % 10;
 int digitoneH = humidity / 10;
```

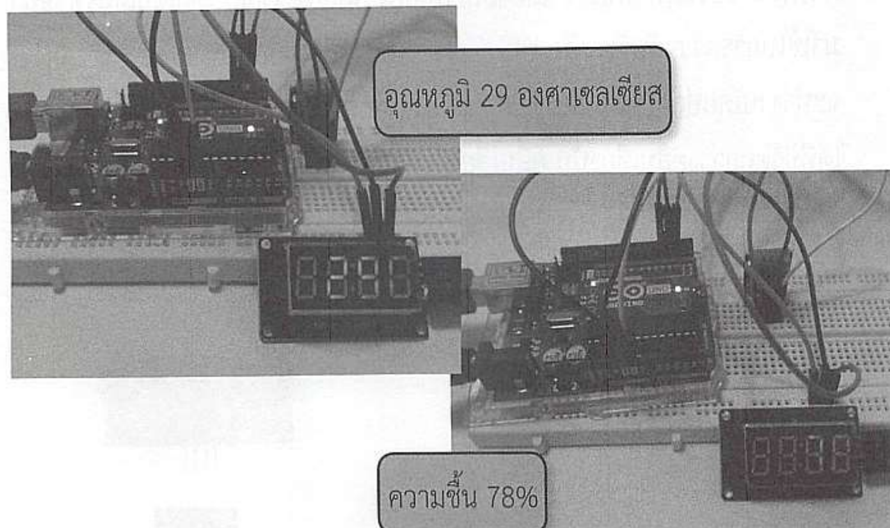
```
int digittwoH = humidity % 10;
 tm1637.display(1,digitoneT);
 tm1637.display(2,digittwoT);
 tm1637.display(3,12); // put a C at the end
 delay (3000);
 tm1637.display(1,25);
 tm1637.display(2,digitoneH);
 tm1637.display(3,digittwoH);
//Wait 3 seconds before accessing sensor again.
//Fastest should be once every two seconds.
 delay(3000);
// end loop()
}
```

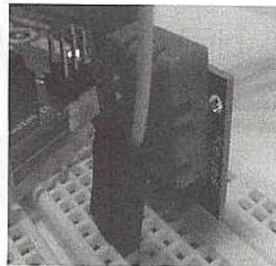


การ Verify (Compile) และการ Upload



เมื่อระบบทำการ Upload Sketch Code เรียบร้อยแล้ว เซ็นเซอร์ตรวจวัดค่าอุณหภูมิและค่าความชื้นจะทำงานทันที โดยเซ็นเซอร์จะทำการตรวจวัดทุก 3 นาที หรือทุกๆ 2 นาที และส่งค่าไปยังบอร์ด Arduino โดยตัวเลขค่าทั้งสอง จะปรากฏที่ LED 4-Digit Display Module และตาม Sketch Code ได้หน่วงเวลาไว้ที่ 3,000 มิลลิวินาที ในการแสดงค่าอุณหภูมิสลับกับค่าความชื้น





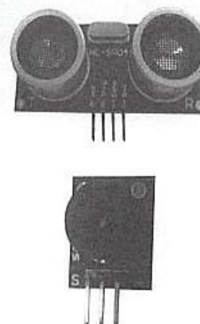




 Detecting Obstacles และ Warning Ultrasonic Buzzer

Detecting Obstacles and Warning Ultrasonic Buzzer คือ โปรเจ็กต์การตรวจจับวัตถุที่กีดขวาง และจะได้รับเสียง Buzzer เตือน โดยเซ็นเซอร์ที่จะนำมาใช้ในการตรวจจับวัตถุ คือ Ultrasonic Sensor Module ซึ่งเซ็นเซอร์ดังกล่าวจะทำงานโดยปล่อยคลื่นเสียงที่มีความถี่สูง (จนหูเราไม่อาจได้ยิน) ออกไป และเมื่อใดที่มีวัตถุขวางคลื่นเสียงนั้น คลื่นเสียงจะสะท้อนกลับมายังตัวเซ็นเซอร์ สำหรับโปรเจ็กต์นี้ นั่นหมายถึง มีวัตถุกีดขวางอยู่ด้านหน้า เช่นนั้นเราจะเขียน Sketch Code ให้ทุกครั้งที่เซ็นเซอร์ตรวจจับเสียงสะท้อนกลับได้ Buzzer Module จะดังเตือนทันที

อุปกรณ์ที่ใช้



1. Arduino Uno R3
2. Ultrasonic Sensor Module
3. Buzzer Module
4. Jumper Wire
5. Breadboard

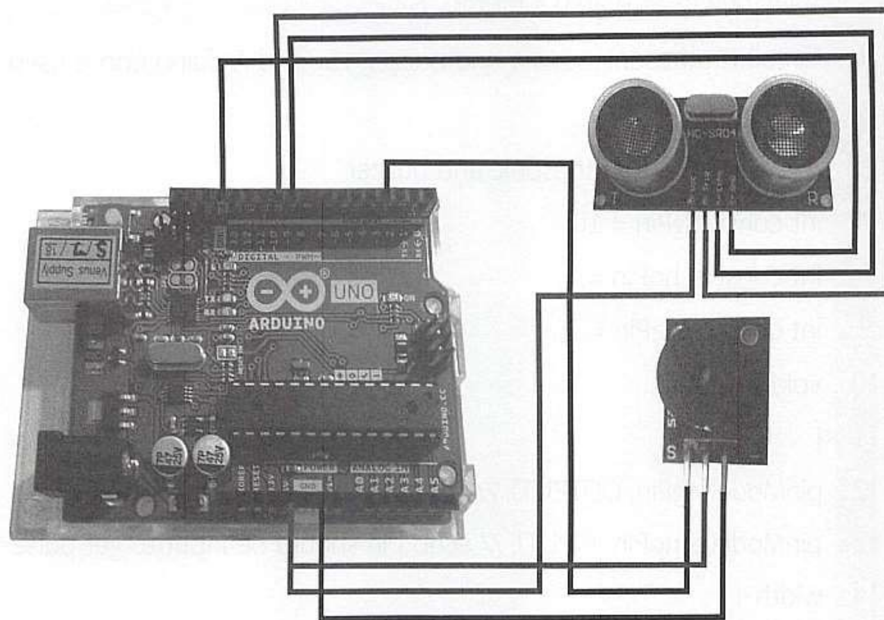
การต่อวงจร

Ultrasonic Sensor Module

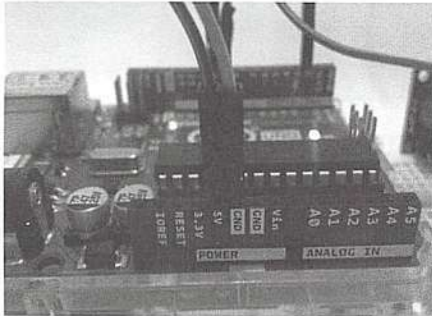
- ขา VCC ของเซ็นเซอร์ต่อกับหมุด 5V บนบอร์ด Arduino
- ขา Trig ของเซ็นเซอร์ต่อกับหมุด ~10 บนบอร์ด Arduino
- ขา Echo ของเซ็นเซอร์ต่อกับหมุด ~9 บนบอร์ด Arduino
- ขา GND ของเซ็นเซอร์ต่อกับหมุด GND บนบอร์ด Arduino

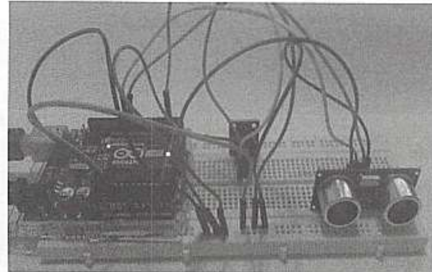
Buzzer Module

- ขา S ของโมดูลต่อกับหมุด D2 บนบอร์ด Arduino
- ขากลางของโมดูลต่อกับหมุด 5V บนบอร์ด Arduino
- ขา - ของโมดูลต่อกับหมุด GND บนบอร์ด Arduino



เมื่อ Ultrasonic Sensor ตรวจจับวัตถุได้ โดยคลื่นเสียงจะสะท้อนวัตถุกลับมายังเซ็นเซอร์ จากนั้นเซ็นเซอร์จะส่งสัญญาณไปยังบอร์ด Arduino และบอร์ดจะส่งสัญญาณไปที่ Buzzer Module





Sketch Code

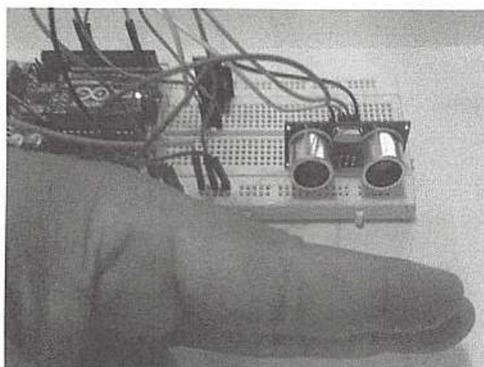
```
/*
This code should work to get warning cross the buzzer when some-
thing be closer than 0.5 meter
Circuit is ultrasonic sensor and buzzer +5v and Arduino uno is used
*/
// Define pins for ultrasonic and buzzer
int const trigPin = 10;
int const echoPin = 9;
int const buzzPin = 2;
void setup()
{
pinMode(trigPin, OUTPUT); // trig pin will have pulses output
pinMode(echoPin, INPUT); // echo pin should be input to get pulse
width
pinMode(buzzPin, OUTPUT); // buzz pin is output to control
buzzering
```

```
}
void loop()
{
// Duration will be the input pulse width and distance will be the
distance to the obstacle in centimeters
int duration, distance;
// Output pulse with 1ms width on trigPin
digitalWrite(trigPin, HIGH);
delay(1);
digitalWrite(trigPin, LOW);
// Measure the pulse input in echo pin
duration = pulseIn(echoPin, HIGH);
// Distance is half the duration devided by 29.1 (from Datasheet)
distance = (duration/2) / 29.1;
// if distance less than 0.5 meter and more than 0 (0 or less
means over range)
if (distance <= 50 && distance >= 0) {
// Buzz
digitalWrite(buzzPin, HIGH);
} else {
// Don't buzz
digitalWrite(buzzPin, LOW);
}
// Waiting 60 ms won't hurt any one
delay(60);
}
```

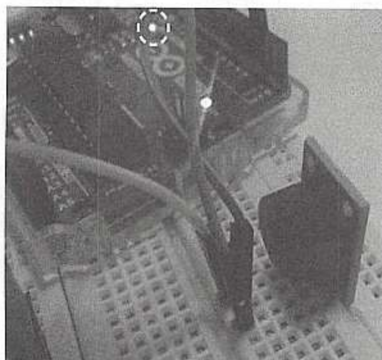
การ Verify (Compile) และการ Upload



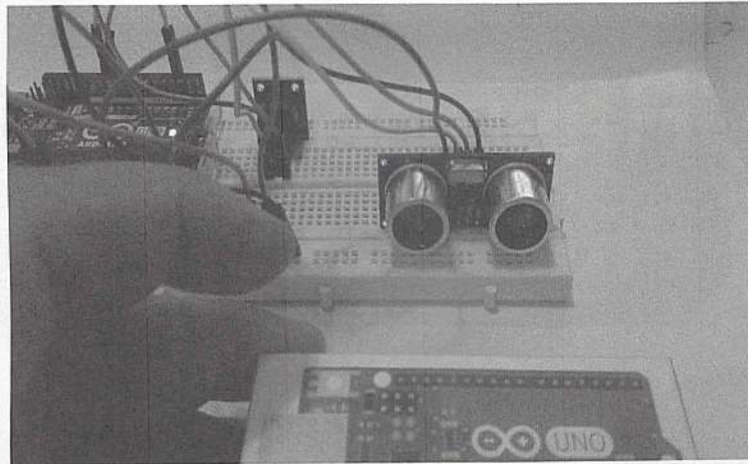
หลังจากที่ Upload Sketch Code ชุด Ultrasonic Sensor Module จะทำงานทันที โดยปล่อยคลื่นเสียงที่มีความถี่สูงออกไป การทดสอบแรกเราจะใช้นิ้วมือวางในตำแหน่งที่เซ็นเซอร์ตรวจจับได้

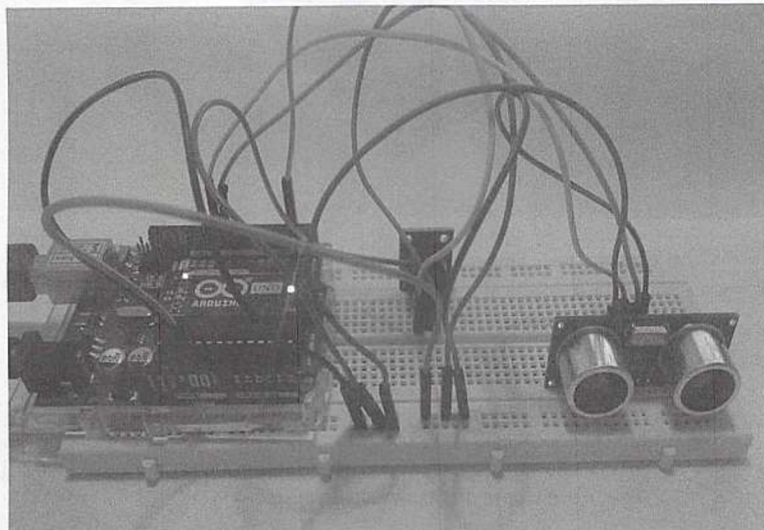


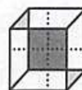
Buzzer Module ดังเตือน พร้อมกับไฟ LED L บนบอร์ด Arduino

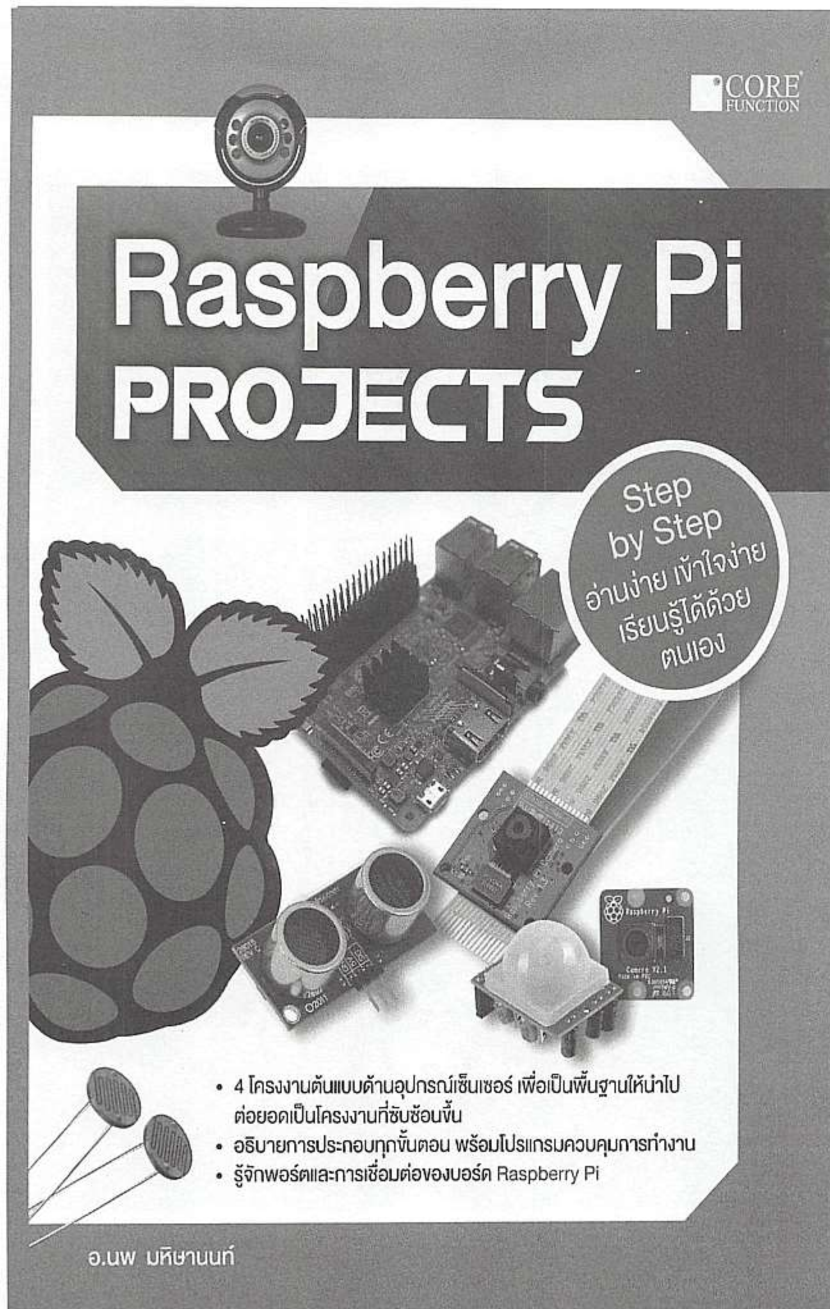


จากนั้นลองเอาวัตถุอย่างอื่น เช่น กล่อง Arduino Uno R3 วางในตำแหน่งที่เซ็นเซอร์ตรวจจับได้



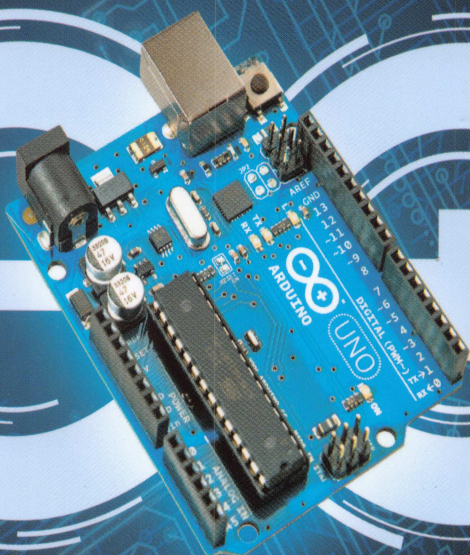





CORE FUNCTION
Raspberry Pi
PROJECTS
Step by Step
อ่านง่าย เข้าใจง่าย
เรียนรู้ได้ด้วยตนเอง
• 4 โครงงานต้นแบบด้านอุปกรณ์เซ็นเซอร์ เพื่อเป็นพื้นฐานให้นำไปต่อยอดเป็นโครงงานที่ซับซ้อนขึ้น
• อธิบายการประกอบทุกขั้นตอน พร้อมโปรแกรมควบคุมการทำงาน
• รู้จักพอร์ตและการเชื่อมต่อของบอร์ด Raspberry Pi
อ.นพ มหิษานนท์

CORE FUNCTION

ARDUINO STARTUP สนุกสุดเหวี่ยงกับเซ็นเซอร์



ถ้าการเรียนรู้บอร์ด Arduino เป็นไปอย่างสนุกสุดมันส์ คุณจะสามารถสนุกสุดเหวี่ยงกับการเขียนโปรแกรมสั่งงาน และประยุกต์ใช้ Arduino ได้แบบไม่รู้จบ

- ไมโครคอนโทรลเลอร์ และ Arduino คืออะไร
- การเขียนภาษา C สำหรับ Arduino
- ติดตั้งและลองใช้ Arduino IDE เขียนโปรแกรม
- Arduino Web Editor เขียนโปรแกรมผ่านบราวเซอร์
- ลอง-เล่น กับเซ็นเซอร์หลากหลายแบบ พร้อมโปรแกรมสั่งงานและการต่อวงจร
- แก้ไขโปรแกรมตัวอย่างที่มี เพื่อต่อยอดเป็นโครงงานที่ซับซ้อนขึ้น

ISBN 978-616-7502-84-7



ราคา 265 บาท

อ.นพ มหิษานนท์